



TUGAS AKHIR - TE 145561

**PERANCANGAN MESIN BOR PCB MENGGUNAKAN
CNC: TATAP MUKA PERGERAKAN MESIN CNC
SECARA MANUAL MENGGUNAKAN VISUAL BASIC**

Aidin Amsyar
NRP 2214030080

Dosen Pembimbing
Fajar Budiman, ST.,M.Sc.

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - TE 145561

**PCB DRILLING MACHINE DESIGN USING CNC:
INTERFACING OF MANUALLY CNC MACHINE
MOVEMENT BY VISUAL BASIC**

Aidin Amsyar
NRP 2214030080

Advisor
Fajar Budiman, ST.,M.Sc.

COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017

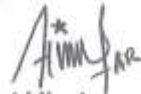
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **"PERANCANGAN MESIN BOR PCB MENGGUNAKAN CNC: TATAP MUKA PERGERAKAN MESIN CNC SECARA MANUAL MENGGUNAKAN VISUAL BASIC"** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 18 Juli 2017



Aidin Amsyar

NRP. 2214 030 080

-----Halaman ini sengaja dikosongkan-----

**PERANCANGAN MESIN BOR PCB MENGGUNAKAN CNC;
TATAP MUKA PERGERAKAN MESIN CNC SECARA
MANUAL MENGGUNAKAN VISUAL BASIC**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada**

**Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing



Fajar Budiman, ST., MSc.
NIP. 1986 07072014 04 1001

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

PERANCANGAN MESIN BOR PCB MENGGUNAKAN CNC: TATAP MUKA PERGERAKAN MESIN CNC SECARA MANUAL MENGGUNAKAN VISUAL BASIC

Nama : Aidin Amsyar
Pembimbing : Fajar Budiman ST., MSc.

ABSTRAK

PCB (Printed Circuit Board) adalah sebuah papan yang dapat digunakan untuk membuat sebuah sirkuit elektronik dari logam yang nantinya akan menghubungkan komponen elektronika yang berbeda jenis maupun yang sejenis tanpa menggunakan kabel sebagai perantara aliran listrik. Untuk menempatkan Komponen elektronika pada sebuah PCB maka diperlukannya pengeboran pada PCB. Pengeboran PCB yang dilakukan secara manual akan mengurangi tingkat kepresisian lubang komponen selain itu untuk melakukan produksi secara banyak maka tingkat kesamaan dari masing-masing jalur yang dibuat akan sangat tinggi. Oleh karena itu, diperlukan sebuah mesin yang dapat melakukan pengeboran secara presisi dan dengan pengoperasian yang sederhana.

Pada Tugas Akhir ini dibuat sebuah *interface* menggunakan sebuah software pemrograman grafis. Software ini digunakan untuk mengendalikan kerja dari mesin CNC untuk mengebor PCB, yang dapat digerakkan secara manual atau dengan pengiriman kode numerik. Dimana ini bisa digunakan untuk setting awal dan kalibrasi CNC bor otomatis.

Pada pengujian motor stepper untuk sumbu X antara perhitungan dan pengukuran memiliki perbedaan antara 0,00 mm – 0,18 mm. Untuk motor stepper pada sumbu Y antara perhitungan dan pengukuran memiliki perbedaan antara 0,02 mm – 0,24 mm. Dan untuk motor stepper pada sumbu X antara perhitungan dan pengukuran memiliki selisih antara 0,01 mm – 0,28 mm. Untuk kedalaman dalam proses pengeboran sedalam 3 mm diukur mulai bagian atas PCB.

Kata Kunci: CNC, *Interface*, Motor Stepper

-----Halaman ini sengaja dikosongkan-----

PCB DRILLING MACHINE DESIGN USING CNC: INTERFACING OF MANUALLY CNC MACHINE MOVEMENT BY VISUAL BASIC

Name : Aidin Amsyar
Advisor : Fajar Budiman ST., MSc.

ABSTRACT

PCB (Printed Circuit Board) is a board that can be used to make an electronic circuit of metal (copper) that will connect different electronic components without using a cable as an intermediary of electricity. To place the Electronic Components on a PCB, Drilling on the PCB is required. Drilling on the PCB manually (human hands) reduces the precision of the part holes. In addition to printing many PCBs for mass production will adequately affect the degree of similarity and standard of each lane created. Therefore, we need a machine that can do precision drilling and has simple operation.

Therefore, this final project it is designed an interface operation panel using a graphics programming software. The software is used to create programs that replace the function of the buttons on the panel. This will be useful for first setting and calibration for an automatic drilling CNC machine.

In the motor stepper testing for the X axis between the calculation and measurement has a difference between 0.00 mm - 0.18 mm. For a motor stepper on the Y axis, the calculations and measurements have a difference between 0.02 mm - 0.24 mm. And for a motor stepper on the X axis, the calculations and measurements have a difference between 0.01 mm - 0.28 mm. For depth in drilling process as deep as 3 mm measured starting on top of PCB. For this small error, it can be used for calibration and setting for automatic CNC machine.

Keyword: CNC, Interface, Motor Stepper

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis haturkan kehadiran Tuhan yang Maha Esa yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada seluruh Utusan Tuhan yang Maha Esa, yang membimbing Umat manusia menjadi pribadi yang lebih baik.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma 3 pada studi komputer kontrol, Departemen Teknik Elektro Otomasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

PERANCANGAN MESIN BOR PCB MENGGUNAKAN CNC: TATAP MUKA PERGERAKAN MESIN CNC SECARA MANUAL MENGGUNAKAN VISUAL BASIC

Dalam Tugas Akhir ini dirancang tatapmuka yang dapat mengendalikan bor yang berada pada mesin CNC secara manual, pergerakan yang dapat dilakukan diantaranya pada sumbu X+,X-, Y+, Y-, Z+, dan Z-. Selain dilakukan secara manual juga dibuat tatap muka secara otomatis yang dikerjakan oleh kelompok reset yang bernama Duviky Erison dengan judul “Perancangan Mesin Bor PCB Menggunakan CNC: Tatap Muka Pergerakan Mesin CNC Secara Otomati Menggunakan Visual Basic”.

Penulis mengucapkan terimakasih kepada kedua orang tua dan keluarga penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Fajar Budiman, ST.,MSc. ats segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terimakasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf ats segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan dikemudian hari.

Surabaya, Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi Penelitian.....	2
1.6 Sistematika Laporan	3
1.7 Relevansi.....	3
BAB II TEORI PENUNJANG.....	5
2.1 <i>Computer Numerical Control</i> (CNC)	5
2.2 Pemrograman Mesin CNC	5
2.3 Motor DC.....	8
2.4 Motor Stepper	9
2.5 <i>Coil Excitation Types</i>	11
2.6 Tahapan Pergerakan Motor Stepper.....	14
2.7 Arduino Uno	15
2.8 Microsoft Visual Basic	15
2.9 EAGLE	17
2.10 PCB G-code	18
BAB III PERANCANGAN PEMBUATAN ALAT	19
3.1 Perancangan Mekanik.....	20
3.2 Perancangan Hardware	26
3.3 Perancangan Software.....	30
3.4 Hasil Pembuatan	42
BAB IV HASIL DAN PEMBAHASAN	45
4.1 Pengujian Akurasi.....	45
4.2 Pengujian Keseluruhan	55

BAB V PENUTUP	63
5.1 Kesimpulan.....	63
5.2 Saran.....	63
DAFTAR PUSTAKA	65
LAMPIRAN	67
RIWAYAT HIDUP	79

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Arah Arus dan Arah Medan Magnet Melewati Konduktor	9
Gambar 2.2 Bagan Jenis-jenis Motor DC.....	9
Gambar 2.3 Bentuk Fisik Motor Stepper.....	10
Gambar 2.4 Susunan Koil Motor Stepper Unipolar	12
Gambar 2.5 Susunan Koil Motor Stepper Bipolar.....	14
Gambar 2.6 Arduino Uno R3	15
Gambar 2.7 Jendela Kerja Visual Basic 2010	16
Gambar 2.8 Tampilan Utama EAGLE	17
Gambar 2.9 Tampilan PCB G-code.....	18
Gambar 3.1 Diagram Fungsional Mesin CNC	19
Gambar 3.2 Mesin CNC Tampak Samping.....	21
Gambar 3.3 Mesin CNC Tampak Depan.....	21
Gambar 3.4 Kerangka Utama CNC	22
Gambar 3.5 Sambungan Antar Frame Pada Kerangka Utama	22
Gambar 3.6 Penopang Besi Pada Sumbu X	23
Gambar 3.7 Bagian Penghubung Antara Sumbu X dan Sumbu Z.....	23
Gambar 3.8 Penggabungan Ball screw Dengan Komponen Z	23
Gambar 3.9 Pemasangan Untuk Sumbu Y	24
Gambar 3.10 Bagian Komponen Untuk Sumbu Z.....	25
Gambar 3.11 Komponen Untuk Menggabungkan Motor Stepper Dengan <i>Ball screw</i>	25
Gambar 3.12 Pin Output Driver A4988	26
Gambar 3.13 Rangkaian Relay	27
Gambar 3.14 Rangkaian Power supply	28
Gambar 3.15 Shield CNC V3.....	29
Gambar 3.16 Konfigurasi Arduino Dengan Shield CNC	29
Gambar 3.17 Koneksi Interface dengan Arduino	30
Gambar 3.18 Script Pengelompokan G-code	31
Gambar 3.19 Flow Chart Pergerakan Motor Stepper	32
Gambar 3.20 Pemrograman Tombol X+	32
Gambar 3.21 Pemrograman Tombol Y+	33
Gambar 3.22 Pemrograman Tombol Z+	34
Gambar 3.23 Pemrograman Tombol X-	35
Gambar 3.24 Pemrograman Tombol Y-	36
Gambar 3.25 Pemrograman Tombol Z-	37
Gambar 3.26 Program Memulai dan Menghentikan Putaran Spindel ...	38
Gambar 3.27 Program Pengembalian ke Posisi Awal	38

Gambar 3.28 Tampilan Setelah Program VB Dijalankan	39
Gambar 3.29 Tampilan ketika Browse File Ditekan.....	39
Gambar 3.30 File yang Digunakan	40
Gambar 3.31 Icon Run ULP	41
Gambar 3.32 Tampilan Awal PCB-G-code	41
Gambar 3.33 Pemberian Nilai Pada Sumbu Z	41
Gambar 3.34 Mesin CNC Tampak Depan	42
Gambar 3.35 Mesin CNC Tampak Samping	42
Gambar 3.36 Tampilan Aplikasi Interface CNC	43
Gambar 4.1 Penempatan Bolpen Pada Mesin CNC	45
Gambar 4.2 Panjang Sumbu X Setelah 20 Step	46
Gambar 4.3 Panjang Sumbu X Setelah 140 Step	47
Gambar 4.4 Panjang Sumbu X Setelah 400 Step	47
Gambar 4.5 Perbandingan Dari Seluruh Data Pada Sumbu X	49
Gambar 4.6 Panjang Sumbu Y Setelah 80 Step	49
Gambar 4.7 Panjang Sumbu Y Setelah 250 Step	50
Gambar 4.8 Panjang Sumbu Y Setelah 350 Step	50
Gambar 4.9 Perbandingan Dari Seluruh Data Pada Sumbu Y	52
Gambar 4.10 Panjang Sumbu Z Setelah 120 Step.....	52
Gambar 4.11 Panjang Sumbu Z Setelah 180 Step.....	53
Gambar 4.12 Panjang Sumbu Z Setelah 500 Step.....	53
Gambar 4.13 Perbandingan Dari Seluruh Data Pada Sumbu Z.....	54
Gambar 4.14 Rangkaian RTC Sebelum Dibor	55
Gambar 4.15 Rangkaian RTC Saat Dilakukan Proses Pengeboran.....	55
Gambar 4.16 Rangkaian RTC Tampak Atas Setelah Dibor.....	56
Gambar 4.17 Grafik Data Hasil Percobaan Rangkaian RTC	57
Gambar 4.18 Rangkaian Driver Relay Tampak Atas Sebelum Dibor....	57
Gambar 4.19 Rangkaian Driver Relay Tampak Atas Setelah Dibor	58
Gambar 4.20 Rangkaian Driver Relay Tampak Bawah Setelah Dibor ..	58
Gambar 4.21 Grafik Data Hasil Percobaan Rangkaian RTC	59
Gambar 4.22 Rangkaian Amplifier Tampak Atas Sebelum Dibor.....	59
Gambar 4.23 Rangkaian Amplifier Tampak Atas Setelah Dibor.....	60
Gambar 4.24 Rangkaian Amplifier Tampak Bawah Setelah Dibor	60
Gambar 4.25 Grafik Data Hasil Percobaan Rangkaian RTC	61

DAFTAR TABEL

	Halaman
Tabel 2.1 Macam-macam G-code	7
Tabel 2.2 Macam-macam M-code	7
Tabel 2.3 Pemberian Tegangan Untuk Operasi Full-Step.....	12
Tabel 2.4 Pemberian Tegangan Untuk Operasi Half-Step	13
Tabel 2.5 Mode Full Step	14
Tabel 2.6 Mode Half Step	14
Tabel 4.1 Data Hasil Pengujian Motor Stepper X.....	48
Tabel 4.2 Data Hasil Pengujian Motor Stepper Y	51
Tabel 4.3 Data Hasil Pengujian Motor Stepper Z	54
Tabel 4.4 Data Pengeboran Rangkaian RTC	56
Tabel 4.5 Data Pengeboran Rangkaian Driver Relay.....	58
Tabel 4.6 Data Pengeboran Rangkaian Amplifier.....	60

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

PCB (*Printed Circuit Board*) adalah sebuah papan yang dapat digunakan untuk membuat sebuah sirkuit elektronik dari logam yang nantinya akan menghubungkan komponen elektronika yang berbeda jenis maupun yang sejenis tanpa menggunakan kabel sebagai perantara aliran listrik. Papan sirkuit ini, sudah diproduksi secara massal dengan cara pencetakan untuk keperluan elektronika yang ada hubungannya dengan kelistrikan. PCB dapat dibedakan menjadi beberapa macam, diantaranya adalah PCB dot, PCB *bread*, dan PCB polos. Untuk PCB polos sebelum dapat digunakan harus melalui proses pencetakan jalur sesuai dengan rangkaian yang diinginkan, kemudian dilakukan proses *etching*, dan kemudian dilakukan proses pelobangan untuk tempat komponen.

Proses pengeboran pada PCB cukup sulit dan kurang efisien apabila dikerjakan oleh manusia, selain hal tersebut pengeboran juga membutuhkan waktu yang cukup lama apalagi dalam suatu jalur kaki untuk masing-masing komponen akan cukup banyak. Dengan didukungnya perkembangan komputer masa kini, komputer telah diaplikasikan ke dalam alat-alat mesin perkakas, yang salah satunya adalah mesin bor.

Hasil perpaduan antara teknologi komputer dan teknologi mekanik inilah yang selanjutnya dinamakan dengan CNC (*Computer Numerically Controlled*). Dikatakan sebagai CNC karena proses penggerakan dari mesin perkakas tersebut sepenuhnya digerakkan menggunakan komputer. Sebelum dapat diproses oleh CNC, komputer membutuhkan kode *G-code* agar hardware pada CNC dapat mengenali perintah yang diberikan oleh komputer melalui kode tersebut.

Dalam pengkodeannya dari bentuk gambar ke bentuk *G-code* membutuhkan suatu konverter yang tersimpan dalam format .gcd yang selanjutnya dapat terbaca baik oleh komputer maupun hardware dari CNC tersebut. Dengan demikian untuk menambah tingkat kepresisian dan kemudahan dalam melakukan suatu proses pengeboran, maka kami membuat mesin pengeboran terhadap PCB menggunakan CNC.

1.2 Permasalahan

Berdasarkan analisa yang dilakukan pada latar belakang, maka masalah yang terjadi di dalam proses pengeboran PCB adalah kurangnya efisiensi dalam pengeboran dan memakan banyak waktu untuk sekali proses pengeboran secara manual. Selain itu tingkat kepresisian untuk mengebor secara manual masih cukup rendah, apalagi apabila sebuah sirkuit tersebut diproduksi secara masal.

1.3 Batasan Masalah

Adapun batasan masalah dari Tugas Akhir ini adalah ukuran PCB (besar sirkuit) yang akan di bor harus kurang dari 10 cm x 15 cm, karena ukuran kerja untuk mesin CNC ini adalah 10 cm x 15 cm x 4,5 cm. selain itu penentuan titik nol ditentukan sesuai ketentuan yang diberikan oleh pembuat.

1.4 Tujuan

Adapun tujuan dibuatnya Tugas Akhir ini adalah untuk membuat mesin bor PCB yang dapat melubangi bagian PCB sesuai dengan sirkuit yang telah ditentukan dengan tingkat kepresisian yang tinggi, menambah tingkat efisiensi terhadap pengeboran PCB, serta mengurangi waktu yang dibutuhkan untuk melubangi sebuah sirkuit elektronik

1.5 Metodologi Penelitian

Pembuatan alat ini dilakukan melalui beberapa tahapan metodologi, yaitu, tahap persiapan, tahap perencanaan dan pembuatan alat, tahap pengujian dan analisis, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap persiapan akan dilakukan pencarian referensi, dan study literatur. Dimana literatur diperoleh dari paper CNC PCB drilling machine. Pada tahap perencanaan dan pembuatan alat, akan dilakukan pembuatan hard ware dari CNC yang dapat bergerak sesuai dengan sumbu X, Y, dan Z. selain itu pada setiap motor stepper yang digunakan, juga akan dibuat driver untuk motor stepper agar dapat dikendalikan melalui arduino. Untuk pengeborannya digunakan motor DC beserta driver motor DC tersebut. Setelah itu akan dilakukan pemrograman dengan menggunakan arduino uno untuk menjalankan mesin CNC dengan menggunakan metode grbl. Setelah dilakukan perencanaan dan pembuatan alat, pengujian yang telah diperoleh selanjutnya akan

dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari Arduino, CNC, motor stepper, dan motor DC

Bab III Perancangan dan Pembuatan Alat

Bab ini membahas tentang penjelasan dari pembuatan *hard ware* dan *soft ware* serta bentuk dari *hard ware* yang telah didesain, serta penjelasan mengenai sirkuit dari driver yang dipergunakan.

Bab IV Pengukuran dan Pengujian

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian tingkat ketepatan dalam pengeboran, efisiensi menggunakan metode grbl untuk menggerakkan mesin CNC dan lain sebagainya. Pada tiap pengujian akan ada analisis terkait metode yang digunakan.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan memberikan manfaat berupa penerapannya pada kalangan masyarakat yang membutuhkan sirkuit elektronik dalam bidang pekerjaannya dan nantinya akan mempermudah apabila akan memproduksi secara masal untuk suatu sirkuit yang dibuat pada PCB.

-----Halaman ini sengaja dikosongkan-----

BAB II

TEORI PENUNJANG

2.1 *Computer Numerical Control (CNC)*

CNC (*Computer Numerical Control*) merupakan suatu mesin yang tergolong dalam mesin perkakas, dikontrol oleh komputer dengan menggunakan bahasa numerik (data perintah dengan kode angka, huruf dan simbol) sesuai dengan standar yang telah disepakati. Sistem kerja dari CNC ini bila dibandingkan dengan mesin perkakas sejenis, maka mesin perkakas CNC lebih teliti, lebih tepat, lebih fleksibel dan lebih cocok apabila melakukan produksi secara masal. Dengan adanya mesin perkakas ini dapat mempermudah dalam produksi yang membutuhkan tingkat kerumitan yang tinggi, selain itu dalam mesin ini juga pengoperasian sepenuhnya ada pada pengendali utama, jadi tidak banyak campur tangan operator selama mesin ini sedang beroperasi.[12]

Numerical Control (NC) adalah suatu format berupa program otomatisasi dimana pergerakan mekanik yang terjadi pada suatu alat permesinan atau peralatan lain dikendalikan oleh suatu program berupa kode angka, angka tersebut berupa data alphanumerical yang menghadirkan suatu instruksi pekerjaan untuk mengoperasikan mesin tersebut. NC bermanfaat untuk produksi suatu item, dimana bentuk, dimensi, rute proses, dan pengerjaan dari suatu mesin, itu bervariasi

Produksi mesin CNC mulai berkembang dengan pesat, yang dipicu dengan perkembangan pada bidang mikrokoprosesor. Hingga sekarang, CNC telah merevolusi proses pencetakan dengan tingkat kepresisian yang sangat bagus dan hasil sama pada setiap item yang diproduksi secara masal.

2.2 *Pemrograman Mesin CNC*

Memprogram mesin CNC merupakan suatu proses memasukan data kekomputer mesin dengan bahasa yang dapat dipahami dan dimengerti oleh CNC. Bahasa pemrograman tersebut berupa bahasa numerik, yaitu bahasa gabungan huruf dan angka. Untuk itu harus dimasukan suatu program ke komputer mesin CNC agar dapat memproses informasi data dan mengubahnya dalam bentuk data dan perintah-perintah gerakan pada mesin CNC.

Untuk dapat menggerakkan mesin CNC diperlukan bahasa pemrograman, berupa kode – kode dalam bentuk huruf dan angka serta metode pemrograman.

2.2.1 Metode Pemrograman

Untuk menunjukkan jalanya pergerakan bor sesuai dengan yang diinginkan, digunakan dua macam metode pemrograman, yaitu pemrograman absolut dan metode pemrograman inkrimental. Yang pertama, pemrograman harga *absolute* merupakan metode pemrograman yang menggunakan satu titik acuan atau satu titik referensi. Dalam menentukan titik koordinat, dari sebuah benda harus sesuai dengan sistim koordinat yang dipakai. Untuk sistim koordinat yang dipakai dalam pemrograman mesin CNC, adalah menggunakan sistim koordinat *cartesius*. [6]

Kemudian yang kedua adalah pemrograman metode *inkrimental*, merupakan suatu metode pemrograman dimana titik referensinya selalu berubah, yaitu titik terakhir yang dituju menjadi titik referensi baru untuk ukuran berikutnya.

2.2.2 Kode-kode Dalam Pemrograman

Bahasa pemrograman *Numerical Control* (NC) adalah program yang terbaca oleh mesin CNC atau yang biasa di kenal dengan *G-code*. Namun dalam kenyataannya, *G-code* ini hanya sebagian dari bahasa pemrograman NC, yang dapat mengendalikan mesin CNC. Kontrol numerik istilahnya diciptakan di Laboratorium Servomechanisms MIT, dan beberapa versi dari NC itu masih dikembangkan secara mandiri oleh pabrik mesin CNC. Versi standar utama yang digunakan di Amerika Serikat telah diselesaikan oleh *Electronic Industries Alliance* di awal 1960-an. Revisi terakhir yang telah disetujui pada bulan Februari 1980 sebagai RS274D. Di Eropa, standar DIN 66.025 / ISO 6.983 sering digunakan sebagai gantinya.

Bahasa kode ini berfungsi sebagai sarana komunikasi antara mesin dengan pemakainya, yakni memberikan informasi data kepada mesin yang harus dipahaminya. Berikut merupakan macam - macam bahasa kode G dan M serta kegunaanya dapat dilihat pada tabel berikut.

Tabel 2.1 Macam-macam *G-code*

Nama kode	Fungsi kode
G00	Gerakan cepat tanpa pemakanan benda kerja (bergerak lurus)
G01	Gerakan memotong/pemakanan benda kerja (bergerak lurus)
G02	Gerakan memotong melingkar searah jarum jam
G03	Gerakan memotong melingkar berlawanan arah jarum jam
G33	Menyayat beberapa jenis ulir dengan kisar konstan
G40	Membatalkan kompensasi radius atau tanpa kompensasi
G41	Kompensasi radius kanan
G42	Perintah kompetensi radius kiri (bubut dalam)
G54	Berarti titik nol benda kerja diaktifkan
G90	Pemrograman absolute
G91	Pemrograman inkremental
G96	Mengatur kecepatan potong.
G97	Pengaturan kecepatan potong konstan <i>OFF</i>
G158	Menentukan awal pemrograman

Untuk program *G-code* pada Tabel 2.1 sebenarnya masih cukup banyak, hanya saja yang paling sering digunakan adalah yang terdapat pada Tabel 2.1. Untuk M-code yang digunakan dalam pemrograman mesin CNC dapat dilihat dalam Tabel 2.2 sebagai berikut :

Tabel 2.2 Macam-macam M-code

Nama Kode	Fungsi Kode
M02	Program berakhir
M03	Menghidupkan poros mesin (<i>spindel on</i>) berputar berlawanan arah jarum jam (CCW).
M04	Spindle ON dengan putaran berlawanan jarum jam
M05	Mematikan poros mesin (<i>spindel off</i>)
M08	Coolant ON
M09	Coolant Off
M30	Langkah terakhir (program end)

Untuk merubah dari sebuah objek gambar menjadi kode-kode yang telah disebutkan diatas maka dibutuhkan sebuah aplikasi konverter yang dapat merubah dari gambar menjadi *G-code*. Banyak sekali aplikasi yang dapat digunakan dalam hal tersebut, format dari file yang telah dikonvert akan mengubah sebuah gambar tersebut baik yang berupa dua dimensi

maupun yang tiga dimensi telah dirubah menjadi kode numerik yang siap dijalankan dalam CNC.

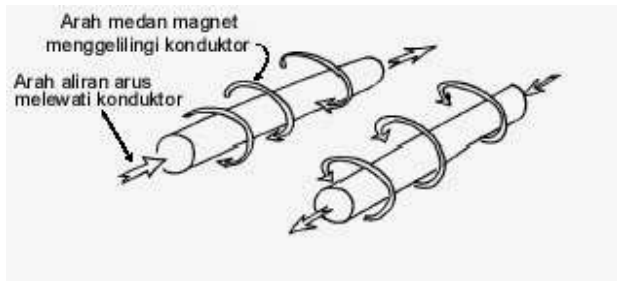
2.3 Motor DC

Motor listrik merupakan peralatan yang dapat mengubah energi listrik menjadi energi mekanik. kebanyakan motor listrik beroperasi melalui interaksi medan magnet dan konduktor pembawa arus untuk menghasilkan putaran. Salah satu motor listrik yang sekarang banyak digunakan adalah motor DC. Motor DC merupakan suatu alat yang dapat mengubah energi listrik searah menjadi energi mekanik putaran. Pada motor DC kumparan medan disebut stator (bagian yang tidak berputar) dan kumparan jangkar yang disebut rotor (bagian yang berputar).[3]

Pada dasarnya prinsip kerja pada motor DC adalah jika ada suatu arus yang melewati suatu konduktor, maka akan timbul medan magnet disekitar konduktor tersebut. medan magnet tersebut hanya berada disekitar konduktor saja saat ada arus yang mengalir. Untuk arah medan magnet ditentukan oleh arah aliran arus pada konduktor.

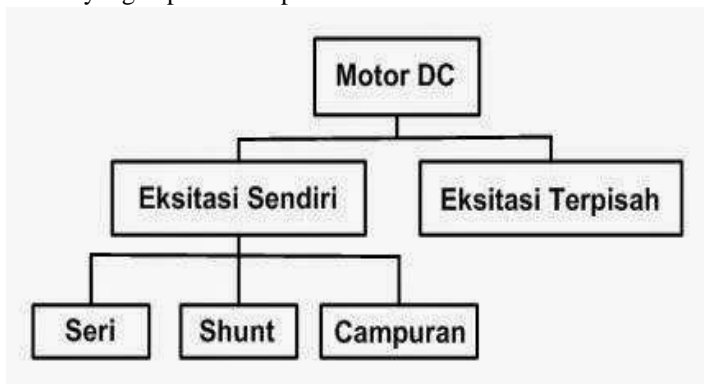
Pada motor DC, daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konversi dari energi listrik menjadi energi mekanik (motor) maupun sebaliknya berlangsung melalui medan magnet, dengan demikian medan magnet disini selain berfungsi sebagai tempat untuk menyimpan energi, sekaligus sebagai tempat berlangsungnya proses perubahan energi.

Untuk menentukan arah putaran motor digunakan kaedah Flamming tangan kiri. Kutub-kutub magnet akan menghasilkan medan magnet dengan arah dari kutub utara ke kutub selatan. Jika medan magnet memotong sebuah kawat penghantar yang dialiri arus searah dengan empat jari, maka akan timbul gerak searah ibu jari. Gaya ini disebut gaya Lorentz, yang besarnya sama dengan F untuk lebih jelasnya dapat dilihat pada Gambar 2.1 mengenai arah arus dan arah medan magnet.



Gambar 2.1 Arah Arus dan Arah Medan Magnet Melewati Konduktor

Menurut sumber yang didapatkan pada motor DC, Motor DC dibedakan menjadi dua, yaitu motor DC eksitasi sendiri dan motor DC eksitasi terpisah. Berikut merupakan diagram dari pembagian jenis-jenis motor DC yang dapat dilihat pada Gambar 2.2.



Gambar 2.2 Bagan Jenis-jenis Motor DC

2.4 Motor Stepper

Motor stepper adalah perangkat elektromekanis yang bekerja dengan memanfaatkan pulsa elektronis menjadi gerakan mekanis diskrit. Pergerakan motor stepper berdasarkan urutan pulsa yang diberikan kepada motor. Oleh karena itu, untuk menggerakkan sebuah motor setep per diperlukan pengendali motor stepper yang dapat membangkitkan pulsa-pulsa periodik. Pada motor stepper juga memiliki torsi penahan, yang memungkinkan menahan posisinya dalam kondisi tertentu. Hal

tersebut sangat berguna untuk digunakan apabila ada suatu sistem yang memerlukan keadaan *start* dan *stop*.

Bila salah satu pin pada motor stepper diberi sumber tegangan, pin tersebut akan mengaktifkan kutub di dalam magnet sebagai kutub utara dan kutub yang tidak diberi tegangan sebagai kutub selatan. Dengan terdapatnya 2 kutub di dalam motor ini, rotor di dalam motor yang memiliki kutub permanen akan mengarah sesuai dengan kutub-kutub input. Kutub utara rotor akan mengarah ke kutub selatan stator sedangkan kutub selatan rotor akan mengarah ke kutub utara stator.[9]

2.4.1 Prinsip Kerja Motor Stepper

Prinsip kerja motor stepper mirip dengan motor DC, sama-sama dicatu dengan tegangan DC untuk memperoleh medan magnet. Bila motor DC memiliki magnet tetap pada stator, motor stepper mempunyai magnet tetap pada rotor. Adapun spesifikasi dari motor stepper adalah banyaknya fasa, besarnya nilai derajat per step, besarnya volt tegangan catu untuk setiap lilitan, dan besarnya arus yang dibutuhkan untuk setiap lilitan.

Motor stepper tidak dapat bergerak sendiri secara kontinyu, tetapi bergerak secara diskrit per-step sesuai dengan spesifikasinya. Untuk bergerak dari satu step ke step berikutnya diperlukan waktu dan menghasilkan torsi yang besar pada kecepatan rendah. Salah satu karakteristik motor stepper yang penting yaitu adanya torsi penahan, yang memungkinkan motor stepper menahan posisinya yang berguna untuk aplikasi motor stepper dalam yang memerlukan keadaan *start* dan *stop*. Untuk mengetahui bentuk fisik dari motor stepper dapat dilihat pada Gambar 2.3



Gambar 2.3 Bentuk Fisik Motor Stepper [8]

2.4.2 Karakteristik Motor Stepper

Pada motor stepper terdapat tiga karakteristik utama diantaranya adalah tegangan, resistansi, dan derajat per step. Untuk lebih jelasnya berikut akan dijelaskan mengenai masing-masing dari karakteristik motor stepper:

a. Tegangan

Tiap motor stepper mempunyai tegangan rata-rata yang tertulis pada tiap unitnya atau tercantum pada datasheet masing-masing motor stepper. Tegangan rata-rata ini harus diperhatikan dengan seksama karena bila melebihi dari tegangan rata-rata akan menimbulkan panas yang menyebabkan kinerja putarannya tidak maksimal atau bahkan motor stepper akan rusak dengan sendirinya

b. Resistansi

Resistansi per lilitan adalah karakteristik yang lain dari motor stepper. Resistansi ini akan menentukan arus yang mengalir, selain itu juga akan mempengaruhi torsi dan kecepatan maksimum pada motor stepper.

c. Derajat Per Step

Besarnya derajat putaran per step adalah parameter terpenting dalam pemilihan motor stepper karena akan menentukan ukuran langkah gerakan yang paling kecil (resolusi). Tiap-tiap motor stepper mempunyai spesifikasi masing-masing, antara lain 0.72° per step, 1.8° per step, 3.6° per step, 7.5° per step, 15° per step, dan bahkan ada yang 90° per step. Dalam pengoperasiannya kita dapat menggunakan 2 prinsip yaitu *full step* atau *half step*. Dengan *full step* berarti motor stepper berputar sesuai dengan spesifikasi derajat per stepnya, sedangkan *half step* berarti motor stepper berputar setengah derajat per step dari spesifikasi motor stepper tersebut.

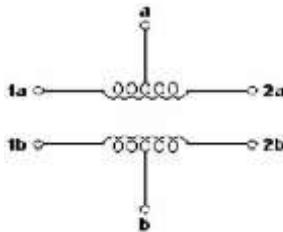
2.5 Coil Excitation Types

Berdasarkan metode perancangan rangkain pengendalinya, motor stepper dapat dibagi menjadi 2 jenis yaitu motor stepper unipolar dan motor stepper bipolar, berikut akan dijelaskan secara terperinci mengenai masing-masing motor.[9]

2.5.1 Motor Stepper Unipolar

Rangkaian pengendali motor stepper unipolar lebih mudah dirancang karena hanya memerlukan satu switch / transistor setiap lilitannya. Motor stepper unipolar sangat mudah untuk dikontrol dengan menggunakan rangkaian *counter* ‘-n’. Motor stepper unipolar mempunyai karakteristik khusus yaitu berupa lilitan center-tapped dan 1 lilitan sebagai common.

Lilitan common akan mencatu tegangan pada *center-tapped* dan sebagai *ground* adalah rangkaian drivernya. Motor stepper unipolar dapat dikenali dengan mengetahui adanya lilitan *center-tapped*. Jumlah phase dan motor stepper adalah dua kali dari jumlah koilnya. Umumnya pada motor stepper unipolar terdapat dua buah koil. Gambar 2.4 merupakan susunan koil motor stepper unipolar.



Gambar 2.4 Susunan Koil Motor Stepper Unipolar

Pada prinsipnya ada dua macam cara kerja motor stepper unipolar, yaitu *full-step* dan *half-step*. Terlihat pada Tabel 2.3 dan Tabel 2.4

Tabel 2.3 Pemberian Tegangan Untuk Operasi *Full-Step*

	Arah putaran searah jarum jam				Arah putaran berlawanan dengan jarum jam			
	1a	1b	2a	2b	1a	1b	2a	2b
1	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0
3	0	0	1	0	0	1	0	0
4	0	0	0	1	1	0	0	0

Pada *full step*, suatu titik pada sebuah kutub magnet di rotor akan kembali mendapat tarikan medan magnet stator pada lilitan yang sama setelah step ke 4., dan berikutnya dapat diberikan lagi mulai dari step 1.

Setiap step, rotor bergerak searah atau berlawanan dengan jarum jam sebesar spesifikasi derajat per step dan motor stepper.

Setiap step hanya menarik sebuah kutub saja. Tegangan '1' adalah menunjukkan logika dalam level Transistor Transistor Logic (TTL). Besar tegangan sesungguhnya diatur dengan spesifikasi motor stepper yang dipakai.

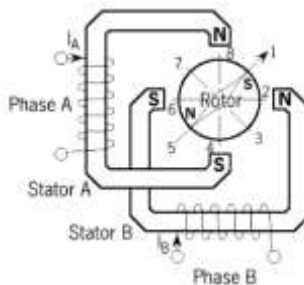
Tabel 2.4 Pemberian Tegangan Untuk Operasi *Half-Step*

	Arah putaran searah jarum jam				Arah putaran berlawanan dengan jarum jam			
	1a	1b	2a	2b	1a	1b	2a	2b
1	1	0	0	0	0	0	0	1
2	1	1	0	0	0	0	1	1
3	0	1	0	0	0	0	1	0
4	0	1	1	0	0	1	1	0
5	0	0	1	0	0	1	0	0
6	0	0	1	1	1	1	0	0
7	0	0	0	1	1	0	0	0
8	1	0	0	1	1	0	0	1

Untuk *half step*, setiap kutub magnet pada rotor akan kembali mendapatkan tarikan dan medan magnet lilitan yang sama setelah step ke 8. berikutnya kembali mulai step 1. Setiap step posisi rotor berubah sebesar setengah derajat dan spesifikasi derajat per step motor stepper.

2.5.2 Motor Stepper Bipolar

Untuk motor stepper dengan lilitan bipolar, diperlukan sinyal pulsa yang berubah-ubah dari positif ke negatif dan sebaliknya. Jadi pada setiap terminal lilitan (A & B) harus dihubungkan dengan sinyal yang mengayun dari positif ke negatif dan sebaliknya. Karena itu dibutuhkan rangkaian pengendali yang agak lebih kompleks daripada rangkaian pengendali untuk motor unipolar. Motor stepper bipolar memiliki keunggulan dibandingkan dengan motor stepper unipolar dalam hal torsi yang lebih besar untuk ukuran yang sama. Gambar 2.5 merupakan susunan koil yang dimiliki oleh motor stepper bipolar.



Gambar 2.5 Susunan Koil Motor Stepper Bipolar

2.6 Tahapan Pergerakan Motor Stepper

Ada dua mode dalam menggerakkan motor stepper yaitu *mode full step* dan *mode half step*. Pada *mode full step* perputaran motor lebih kasar dibandingkan dengan *mode half step*. Ini dikarenakan pada *mode half step* untuk menggerakkan satu step dibutuhkan dua kondisi sehingga perputaran lebih halus, sedangkan pada *mode full step* torsiya lebih besar dibandingkan dengan *mode half step*. Pada Tabel 2.5 dan Tabel 2.6 merupakan *input* untuk *mode full step* dan *mode half step*. [4]

Tabel 2.5 Mode Full Step

	Input 1	Input 2	Input 3	Input 4
NA	0	1	1	1
NB	1	0	1	1
NC	1	1	0	1
ND	1	1	1	0

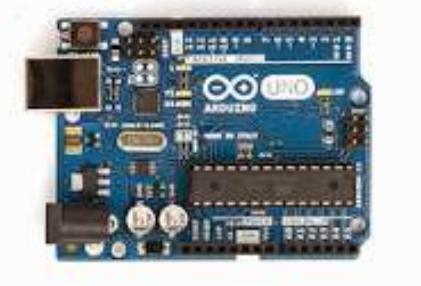
Tabel 2.6 Mode Half Step

	Input 1	Input 2	Input 3	Input 4
NA	0	1	1	1
NB	0	0	1	1
NC	1	0	1	1
ND	1	0	0	1
NA	1	1	0	1
NB	1	1	0	0
NC	1	1	1	0
ND	0	1	1	0

2.7 Arduino Uno

Arduino adalah pengendali *mikro single-board* yang bersifat open-source, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR. Bahasa pemrograman yang dipakai dalam Arduino adalah bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino.

Arduino Uno R3 (Revisi 3), menggunakan ATMEGA328 sebagai microcontrollernya, memiliki 14 pin I/O digital dan 6 pin input analog. Untuk pemrograman cukup menggunakan koneksi USB type A to type B, sama seperti yang digunakan pada USB printer. Untuk lebih jelasnya dapat dilihat pada Gambar 2.6 mengenai bentuk fisik dari arduino uno V3.



Gambar 2.6 Arduino Uno R3 [2]

2.8 Microsoft Visual Basic

Microsoft Visual Basic (sering disingkat sebagai VB) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (COM). Beberapa bahasa skrip seperti *Visual Basic for Applications* (VBA) dan *Visual Basic Scripting Edition* (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda. Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh Microsoft Visual Basic.[11]

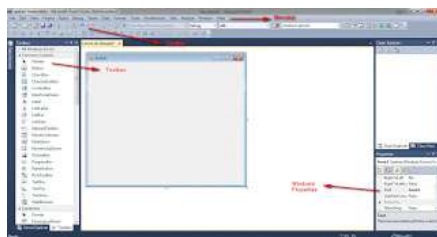
Saat microsoft meluncurkan Microsoft QuickBasic dan Microsoft Basic (dikenal juga sebagai Basic Compiler). Popularitas dan pemakaian BASIC yang luas dengan berbagai jenis komputer turut berperan dalam mengembangkan dan memperbaiki bahasa itu sendiri, dan akhirnya

berujung pada lahirnya Visual Basic yang berbasis GUI (*Graphic User Interface*) bersamaan dengan Microsoft Windows. Pemrograman Visual Basic begitu mudah karena menghemat waktu pemrograman dengan tersedianya komponen-komponen yang siap pakai. Hingga akhirnya Visual Basic juga telah berkembang menjadi beberapa versi, sampai yang terbaru, yaitu Visual Basic 2010.

Jendela Visual Basic atau sering juga disebut lingkungan kerja Visual Basic mempunyai tampilan yang hampir sama dengan tampilan jendela program aplikasi windows yang sudah biasa kita kenal. Di aplikasi visual basic terdapat menu-menu dan toolbar yang memuat icon-icon dan tombol-tombol untuk menjalankan perintah-perintah. Perbedaannya, Visual Basic mempunyai beberapa tambahan komponen, yaitu *Toolbox*, *Windows Project*, dan *Windows Properties*. Berikut akan dijelaskan mengenai masing-masing dari komponen pada visual basic.

Yang pertama adalah *toolbox*, merupakan komponen lingkungan kerja VB yang berisikan tool-tool untuk ditempatkan di form. Jika kita membuat sebuah aplikasi, maka komponen-komponen tersebut akan kita tempatkan di form dan menjadi komponen jendela program. Toolbox ditempatkan disebelah kiri jendela kerja pada *Visual Basic*.

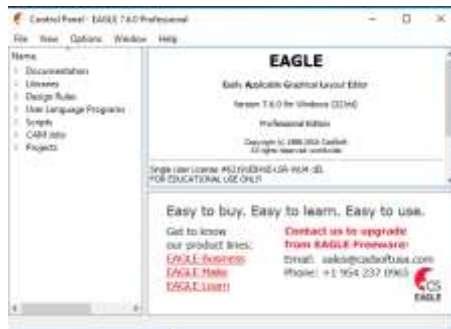
Kemudian yang kedua adalah *windows project*, yang berfungsi untuk menampilkan daftar form dan modul yang terdapat di project aplikasi yang sedang dikerjakan. Letak dari *windows project* ini disebelah kanan jendela kerja Visual Basic. Dan yang terakhir adalah *windows properties*, berfungsi untuk menampilkan daftar properti dari sebuah komponen yang sedang aktif. Kita dapat mengubah properti dari sebuah komponen dengan cara mengaktifkan (mengklik/memilih) komponen tersebut, kemudian mengubah nilai propertinya di *window properties*. Pada Gambar 2.7 merupakan tampilan awal saat visual basic dijalankan.



Gambar 2.7 Jendela Kerja Visual Basic 2010

2.9 EAGLE

EAGLE (*Easily Applicable Graphical Layout Editor*), merupakan sebuah aplikasi gratis untuk mendesain skematik Elektronika yang kemudian diterapkan pada PCB (*Printed Circuit Board*). Dalam perkembangan aplikasi EAGLE sendiri telah memiliki beberapa versi dari aplikasinya, hingga saat ini versi terbaru yang dikeluarkan oleh EAGLE adalah EAGLE 7.8.0. Layar kerja pada eagle dapat dilihat seperti pada Gambar 2.8.

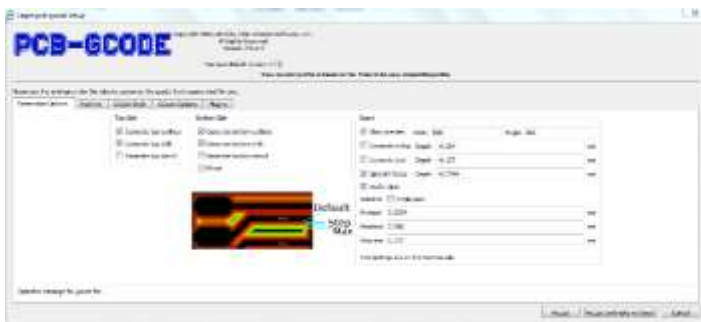


Gambar 2.8 Tampilan Utama EAGLE

Dalam EAGLE memiliki beberapa fitur penting yang dapat memudahkan kita dalam melakukan desain sirkuit elektronik, diantaranya *Schematic editor*, berfungsi untuk menggambar dan mengedit rangkaian skematik elektronika. Di sini akan menambahkan komponen-komponen yang dibutuhkan telah tersedia pada library EAGLE. Kemudian ada *Library Tool*, berfungsi untuk menambahkan berbagai komponen elektronika ke dalam *schematic editor*. *Library tool* ini terdapat berbagai macam komponen yang biasa digunakan dalam bidang elektronik, mulai dari Komponen Diskret (Resistor, kapasitor, dioda) sampai IC Mikrokontroler dan Mikroprosesor dan masih banyak lagi. Kemudian fitur yang terakhir adalah *Eagle Board*, Berfungsi untuk mentransfer rangkaian skematik ke rangkaian PCB. Di sini dapat menata komponen agar sesuai dengan yang diinginkan dan juga melakukan *routing* jalur PCB.

2.10 PCB G-code

PCB *G-code* merupakan sebuah sub menu dari aplikasi EAGLE. PCB *G-code*, ini digunakan untuk mengubah schematic yang telah dibuat menjadi kode numerik yang dapat dikenali oleh mesin CNC (*G-code*). Dalam pengaplikasiannya aplikasi ini memanfaatkan hasil routing yang sebelumnya telah dibuat, dan setelah dilakukan proses ekspor pada PCB *G-code* hasil dari proses tersebut nantinya terdapat dua bagian yaitu bagian bawah (jalur antas komponen) dan bagian atas (peletakan komponen) menjadi kode numerik. Setelah dilakukan konvert dari eagle maka tampilan untuk setting menjadi *G-code* akan tampak seperti pada Gambar 2.9.



Gambar 2.9 Tampilan PCB *G-code*

BAB III

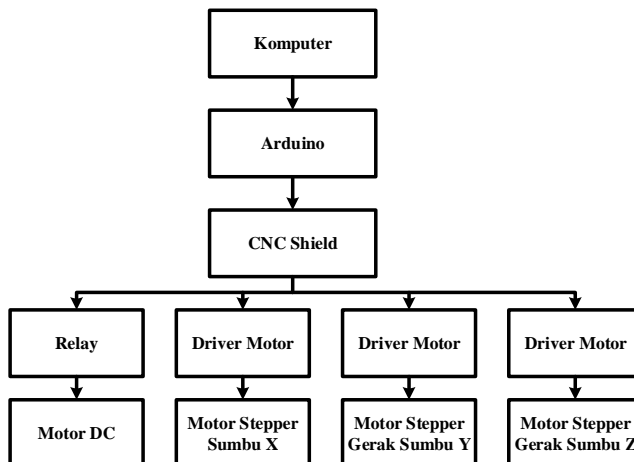
PERANCANGAN ALAT

Pada bab ini akan membahas tentang tahapan yang dilakukan terhadap perancangan dan pembuatan Tugas Akhir Perancangan Mesin Bor PCB Menggunakan CNC. Bagian awal dari bab ini akan dibahas mengenai diagram fungsional dari Tugas Akhir kemudian akan dijelaskan mengenai perancangan hardware yang digunakan, dilanjutkan dengan perancangan software, dan pada bagian terakhir akan dijelaskan mengenai perancangan mekanik yang digunakan.

Pada Gambar 3.1 merupakan diagram fungsional untuk proses kerja dari Mesin Bor PCB Menggunakan CNC

Penjelasan diagram fungsional :

1. Komputer, digunakan untuk menampilkan *interface* dan juga digunakan untuk melakukan konvert dari skematik rangkaian menjadi *G-code*.
2. Arduino Uno, mikrokontroler yang berfungsi sebagai pengendali, pengolah sinyal masuk dan keluar. yang nantinya akan memberikan input kepada driver motor
3. Driver Motor, digunakan untuk menerjemahkan input yang diterima dari Arduino uno sehingga nantinya dapat mengontrol gerakan motor



Gambar 3.1 Diagram Fungsional Mesin CNC

4. Gerak sumbu x, Pergerakan motor stepper pada CNC untuk bagian sudut X
5. Gerak sumbu Y, Pergerakan motor stepper pada CNC untuk bagian sudut Y
6. Gerak sumbu Z, Pergerakan motor stepper pada CNC untuk bagian sudut Z
7. Relay, berfungsi untuk mengaktifkan motor DC dengan masukan yang diterima dari arduino
8. Motor DC, digunakan sebagai spindle untuk proses pengeboran

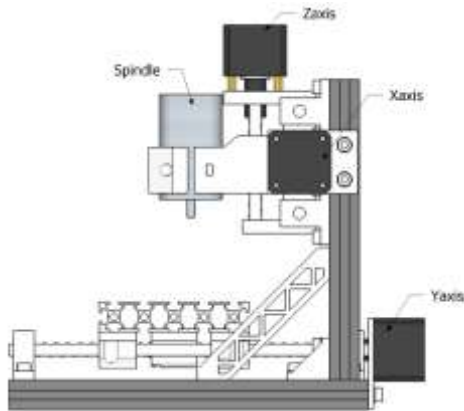
Secara umum sistem yang terdapat pada Gambar 3.1 adalah sistem kerja dari peralatan CNC untuk peneboran PCB. Sebelum dilakukan proses pengeboran, terlebih dahulu harus mendapatkan kode numerik (*G-code*) agar mesin CNC dapat bekerja sesuai dengan kode yang telah diberikan. *G-code* disini akan terbaca melalui *interface* yang ada pada laptop. Kemudian untuk pergerakan pada CNC digunakan metode grbl yang telah di upload pada arduino. Arduino terkoneksi dengan CNC shield dimana dalam CNC shield tersebut terdapat 3 driver motor yang masing-masing menggerakkan untuk sumbu X, sumbu Y, dan Sumbu Z. Dalam CNC shield juga dihubungkan dengan relay agar motor DC sebagai alat pengebor dapat bekerja.

3.1 Perancangan Mekanik

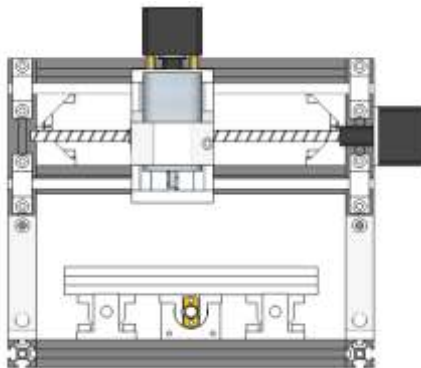
Dalam perancangan mekanik, terdapat beberapa bagian yang harus dibuat untuk dapat bekerja dengan baik. Pada CNC terdapat beberapa bagian yaitu penggerak pada sumbu X, penggerak pada sumbu Y, penggerak pada sumbu Z, dan tempat untuk meletakkan motor DC yang digunakan untuk pengeboran.

3.1.1 Mesin CNC

Setelah semua komponen telah terpasang maka untuk bentuk keseluruhan dari CNC ini memiliki panjang 310 mm, lebar 285 mm, dan tinggi 270 mm. Ukuran tersebut telah diserakan dengan motor stepper yang telah terpasang. Pada Gambar 3.2 dan Gambar 3.3 merupakan gambaran dari mesin CNC yang digunakan.



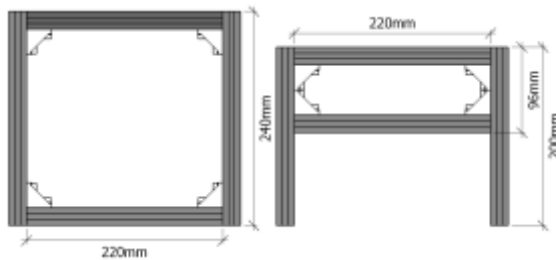
Gambar 3.2 Mesin CNC Tampak Samping



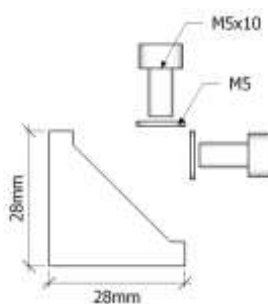
Gambar 3.3 Mesin CNC Tampak Depan

3.1.2 Kerangka Utama CNC

Pada bagian ini kerangka terbuat dari *aluminium profile*, dimana terdapat 3 ukuran yang berbeda. Yang pertama panjangnya 20 cm sejumlah 2 buah, bagian ini nantinya yang akan menopang untuk sumbu Z. Kemudian yang kedua memiliki panjang 22 cm sebanyak 4 buah, bagian ini nantinya yang akan menopang untuk sumbu X. Dan yang terakhir memiliki panjang 24 cm sebanyak 2 buah, bagian ini nantinya digunakan untuk menopang pada sumbu Y.



Gambar 3.4 Kerangka Utama CNC

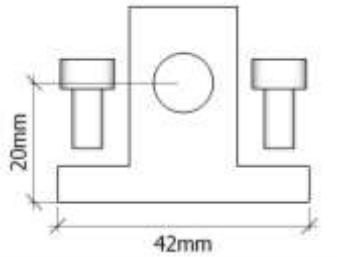


Gambar 3.5 Sambungan Antar Frame Pada Kerangka Utama

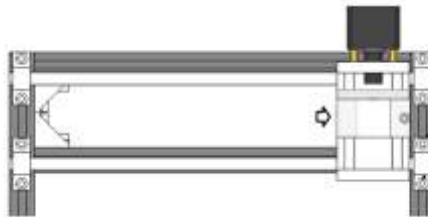
Untuk menggabungkan masing-masing frame digunakan sambungan disetiap sudutnya, untuk sambungan yang digunakan berbentuk siku dengan panjang 28 mm (dapat dilihat pada Gambar 3.5). Dan kemudian diberi baut dengan berdiameter baut 5 mm, serta cincin bautnya.

3.1.3 Pemasangan Untuk Sumbu X

Untuk sumbu X terletak pada bagian tinggi dari kerangka utama CNC, pada sumbu X ini terdapat 2 besi dengan panjang 225 mm, selain itu juga terdapat *ball screw* yang juga memiliki panjang yang sama dengan panjang batang besi. Kedua batang besi dan sebuah *ball screw* ini digunakan untuk menggerakkan tempat pengeboran (seluruh komponen pada sumbu Z). Untuk penopangnya masing-masing besi disambungkan dengan komponen yang tampak pada Gambar 3.6 Dimana komponen tersebut akan menghubungkan dengan frame utama pada CNC.



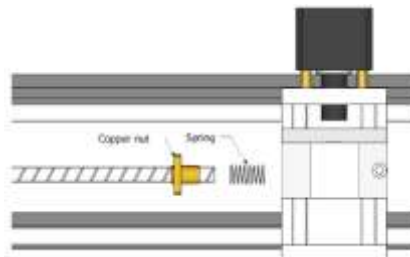
Gambar 3.6 Penopang Besi Pada Sumbu X



Gambar 3.7 Bagian Penghubung Antara Sumbu X dan Sumbu Z

Setelah Terpasang dan tergabung dengan komponen pada sumbu Z, maka bentuknya akan seperti yang tampak pada Gambar 3.7.

Untuk pemasangan *ball screw* yang nantinya akan tersambung dengan motor stepper pada sumbu X dan juga tergabung dengan komponen pada sumbu Z, terdapat sebuah *spring* yang digunakan untuk mengurangi tekanan saat perputaran berlangsung (agar kontak yang terjadi antara komponen sumbu Z dan *ball screw* tidak terlalu besar) dan juga terdapat Chopper nut yang digunakan sebagai penahan dari *spring*. Untuk cara memasangnya dapat dilihat seperti pada Gambar 3.8



Gambar 3.8 Penggabungan *Ball screw* Dengan Komponen Z

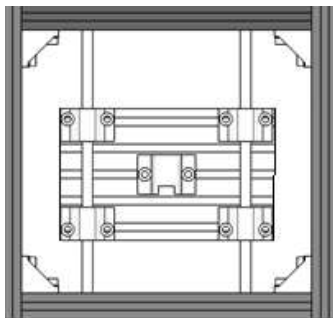
Di masing-masing ujung dari *ball screw* nantinya yang satu akan terhubung dengan motor stepper dan sisi yang lainnya akan terhubung dengan bearing yang juga berfungsi sebagai penggerak dari komponen sumbu Z. Untuk sumbu bagian X ini digunakan motor stepper dengan spesifikasi besar drajat setiap stepnya adalah 1.8° , dengan tegangan 12V - 24V, besar arus 1.33 A, dan tipe dari motor ini adalah tipe *hybrid*.

3.1.4 Pemasangan Untuk Sumbu Y

Dalam sumbu Y ini, saat sistem telah dijalankan yang akan bergerak adalah dudukan pada mesin CNC. Dudukan ini merupakan tempat dari bahan yang akan menjalani proses pengeboran, oleh karena itu antara penggerak pada sumbu Y ini tergabung dengan dudukan pada CNC. Untuk komponen dalam pemasangan sumbu Y ini hampir sama dengan pemasangan pada sumbu X, hanya saja panjang dari batang besi yang digunakan dan *ball screw*nya berbeda.

Pada sumbu Y panjang besi yang digunakan adalah 230 mm sebanyak dua buah, untuk panjang *ball screw* yang digunakan juga sama dengan panjang batang besinya. Untuk lebih jelasnya berikut adalah pemasangan antara dudukan dengan batang besi serta *ball screw*, untuk lebih jelasnya dapat dilihat pada Gambar 3.9.

Dudukan yang digunakan tersebut memiliki lebar 100 mm dan panjang 170 mm. Pada bagian tengah dari dudukan ini juga diberikan *spring*, seperti halnya yang ada pada sumbu X. Untuk sumbu bagian Y ini digunakan motor stepper dengan spesifikasi besar drajat setiap stepnya adalah 1.8° , dengan tegangan 12V - 24V, besar arus 1.33 A, dan tipe dari motor ini adalah tipe *hybrid*.



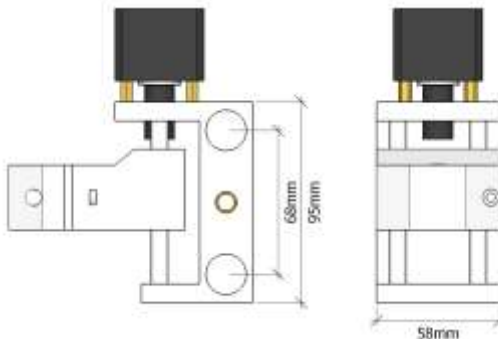
Gambar 3.9 Pemasangan Untuk Sumbu Y

3.1.5 Pemasangan Untuk Sumbu Z

Bahan utama bagian sumbu Z ini terbuat dari Atom yang cukup tebal dan beberapa besi sebagai penyangganya. Pada sumbu Z memiliki lebar komponen sebesar 58 mm dan tinggi 95 mm, selain itu terdapat untuk meletakkan motor DC (sebagai *Spindel*) yang panjangnya 80 mm dan diameter motor DC yang dapat digunakan adalah sekitar 43 mm.

Untuk dapat menggerakkan keatas dan kebawah tempat bornya, terdapat sebuah *ball screw* 90 mm yang diapit oleh batang besi dengan panjang yang sama. Baik dari *ball screw* maupun batang besi memiliki ukuran diameter yang sama seperti yang tampak pada Gambar 3.10. Untuk penggabungan antara motor stepper dengan *ball screw* menggunakan komponen seperti yang tampak pada Gambar 3.11.

Untuk sumbu bagian Z ini digunakan motor stepper dengan spesifikasi besar drajat setiap stepnya adalah 1.8° , dengan tegangan 12V - 24V, besar arus 1.33 A, dan tipe dari motor ini adalah tipe *hybrid*. Untuk pergerakan sumbu Z (naik dan turun) apabila telah dipasang motor DC maksimal hanya dapat bekerja dalam rentang 0 samapi 45 mm.



Gambar 3.10 Bagian Komponen Untuk Sumbu Z



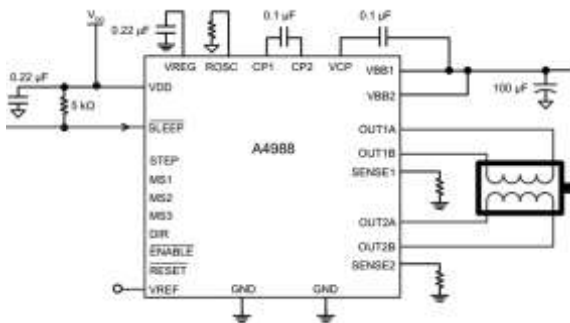
Gambar 3.11 Komponen Untuk Menggabungkan Motor Stepper Dengan *Ball screw*

3.2 Perancangan Hardware

Dalam perangkat elektronik, terdapat beberapa elemen yang harus disusun untuk dapat menggerakkan CNC dengan baik. Elemen-elemen tersebut adalah sebagai berikut:

3.2.1 Driver motor A4988

Driver motor digunakan untuk mengendalikan arah motor, Driver motor jenis A4988 ini digunakan khusus untuk menggerakkan motor stepper dengan metode grbl. Karena pada driver ini pin output yang terhubung pada arduino hanya sebanyak 2, padahal pada motor stepper terdapat 4 kumparan yang masing-masing digunakan untuk mengatur arah gerak serta banyak step yang dikerjakan untuk lebih jelasnya dapat dilihat pada Gambar 3.12.

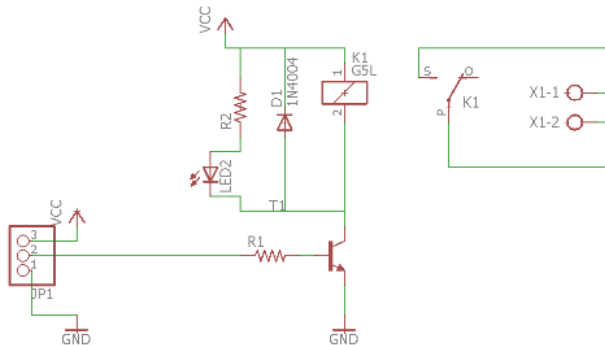


Gambar 3.12 Pin Output Driver A4998

3.2.2 Rangkaian Relay

Relay merupakan suatu piranti elektronik yang bekerja berdasarkan elektromagnetik untuk menggerakkan sejumlah kontektor yang tersusun atau sebuah saklar elektronis yang dapat dikendalikan dari rangkaian elektronik lainnya dengan memanfaatkan tenaga listrik sebagai sumber energinya. Pada rangkaian relay yang telah dibuat, digunakan led sebagai indikator saat kontaktor pada relay bekerja (dari NO menjadi NC).

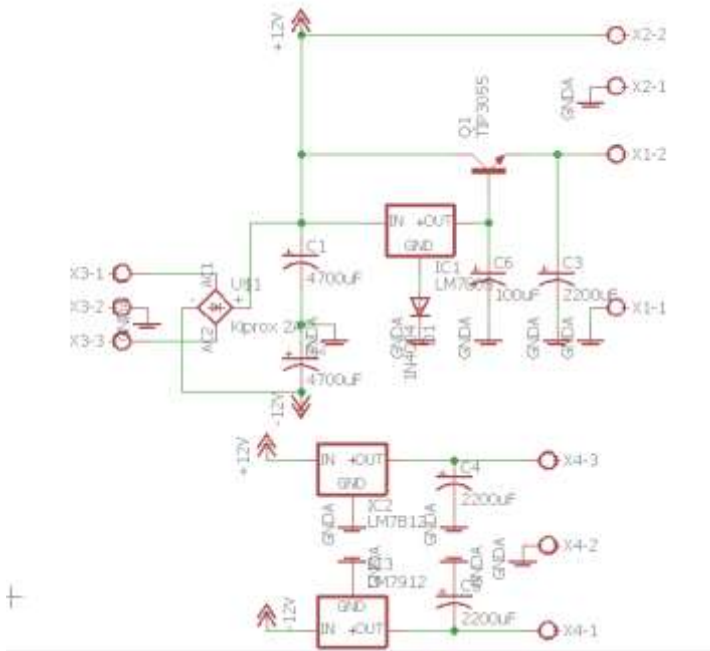
Juga digunakan dioda serta transistor PNP sebagai pengaman pada relay. Untuk pengaktifan relay disambungkan ke pin output arduino, sehingga output pada arduino memberikan tegangan sebesar 5V dan relay akan aktif untuk menyalakan motor DC dengan tegangan 12V. Untuk lebih jelas mengenai rangkaian relay dapat dilihat pada Gambar 3.13 .



Gambar 3.13 Rangkaian Relay

3.2.3 Rangkaian *Power supply*

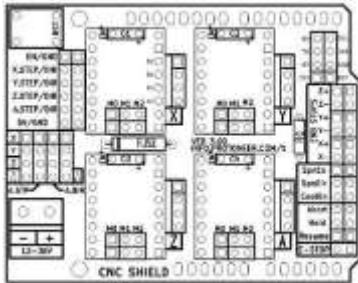
Power supply merupakan suatu hardware komponen elektronika yang berfungsi sebagai suplayer arus listrik dengan terlebih dahulu merubah tegangan AC (*Alternating Current*) menjadi tegangan DC (*Direct Current*). Pada tugasakhir ini digunakan *power supply* sebagai sumber dari motor DC yang digunakan saat proses pengeboran. Dari rangkaian *power supply* ini, digunakan penurun tegangan dari 220V AC menjadi 12V DC, dimana 12V DC ini digunakan pada motor DC. Proses penurunan tegangan tersebut terjadi karena terdapat IC penurun tegangan dengan tipe IC 7812, serta beberapa kapasitor untuk mengurangi nois pada tegangan DC yang dihasilkan. Untuk trafo yang digunakan berjenis CT dengan arus sebesar 2A. Gambar 3.14 merupakan rangkaian dari *power supply*.



Gambar 3.14 Rangkaian *Power supply*

3.2.4 Shield CNC V3

Shield merupakan komponen tambahan untuk mempermudah pengguna arduino dalam beberapa kasus tertentu. Shield yang dimiliki arduino memiliki banyak macam, dalam Tugas Akhir ini shield yang digunakan adalah shield CNC V3. Dalam shield ini tersedia 4 axis, hanya saja yang digunakan Cuma 3 axis yaitu untuk sudut X, sudut Y, dan sudut Z. Untuk tampilan dari shield CNC dapat dilihat pada Gambar 3.15.

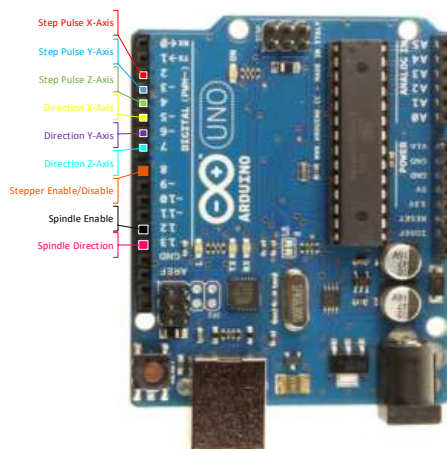


Gambar 3.15 Shield CNC V3

Untuk driver yang dapat digunakan dalam shield ini adalah driver motor A4988. Selain itu dalam shield ini juga telah ada pin *spindel* enable dan direction yang nantinya akan tersambung langsung dengan relay sebelum masuk ke motor DC. Untuk tegangan yang dapat diterima oleh shield ini adalah sekitar 12V – 36V DC.

3.2.5 Koneksi Arduino Dengan Shield CNC

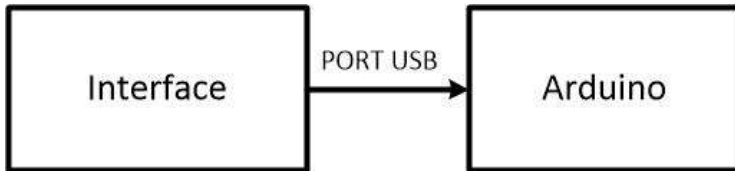
Untuk koneksi antara arduino uno dengan shield CNC, memanfaatkan beberapa pin yang ada pada arduino dan langsung terkoneksi dengan shield CNC. Berikut merupakan konfigurasi dari masing-masing pin yang terkoneksi dalam arduino dapat dilihat pada Gambar 3.16..



Gambar 3.16 Konfigurasi Arduino Dengan Shield CNC

3.2.6 Rangkaian Interface

Interface ini dibuat dari aplikasi microsoft visual basic. Dimana proses dari *interface* ini nantinya digunakan sebagai pengendali CNC, pemasukan program *G-code*, dan koneksi dengan arduino. Sehingga untuk *interface* ini memanfaatkan data serial pada arduino melalui USB pada komputer, seperti yang tampak pada Gambar 3.17.



Gambar 3.17 Koneksi *Interface* dengan Arduino

3.3 Perancangan Software

Dalam perangkat lunak, terdapat beberapa program yang harus dibuat untuk dapat menggerakkan CNC dengan baik. Tahapan pembuatan tersebut adalah sebagai berikut:

3.3.1 Program Penggerak CNC dengan Metode GRBL

Grbl merupakan sebuah metode yang digunakan untuk pemrograman sebuah mesin CNC. Dalam grbl ini telah diatur beberapa kegunaan serta mengindikasikan komponen-komponen yang ada pada mesin CNC, sehingga dengan adanya program grbl ini mikrokontroler dapat mengenali komponen yang digunakan pada mesin CNC.

Dalam grbl sendiri memiliki beberapa sub dari program yang berperan penting dalam menjalankan mesin CNC. Hanya saja dalam hal ini tidak ditampilkan secara keseluruhan programnya, mungkin beberapa bagian inti yang digunakan dalam mesin CNC yang telah dibuat.

1. Config

Dalam header Config.h ini merupakan kumpulan dari keseluruhan program yang ada pada pemrograman CNC menggunakan metode grbl, seperti pengaktifan *spindel*, penentuan boudrate yang akan digunakan, pengaktifan soft limit, dan masih banyak lagi. Namun tidak semua program yang ada telah aktif, oleh karena itu dalam program config ini yang menentukan komponen-komponen tambahan yang akan digunakan. Untuk lebih jelasnya program dapat dilihat pada lampiran A

2. G-code

Untuk bagian ini, pemrograman digunakan untuk mengenalkan kode-kode numerik kepada arduino, seperti G00, M03, M05 dan masih banyak lagi tentunya. Untuk lebih jelasnya dapat dilihat pada Gambar 3.18 Yang merupakan sedikit dari keseluruhan program yang ada.

```
#define MODAL_GROUP_G0 0 //  
[G4,G10,G28,G28.1,G30,G30.1,G53,G92,G92.1] Non-modal  
#define MODAL_GROUP_G1 1 //  
[G0,G1,G2,G3,G38.2,G38.3,G38.4,G38.5,G80] Motion  
#define MODAL_GROUP_G2 2 // [G17,G18,G19] Plane selection  
#define MODAL_GROUP_G3 3 // [G90,G91] Distance mode  
#define MODAL_GROUP_G4 4 // [G91.1] Arc IJK distance mode  
#define MODAL_GROUP_G5 5 // [G93,G94] Feed rate mode  
#define MODAL_GROUP_G6 6 // [G20,G21] Units  
#define MODAL_GROUP_G7 7 // [G40] Cutter radius compensation  
mode. G41/42 NOT SUPPORTED.
```

Gambar 3.18 *Script* Pengelompokan G-code

Seperti yang tampak pada Gambar 3.18 Beberapa kode numerik dikelompokkan menurut pergerakan dan proses, kode yang memiliki proses yang hampir sama dikelompokkan dalam satu grup dan begitu juga dengan yang lainnya.

3. Serial

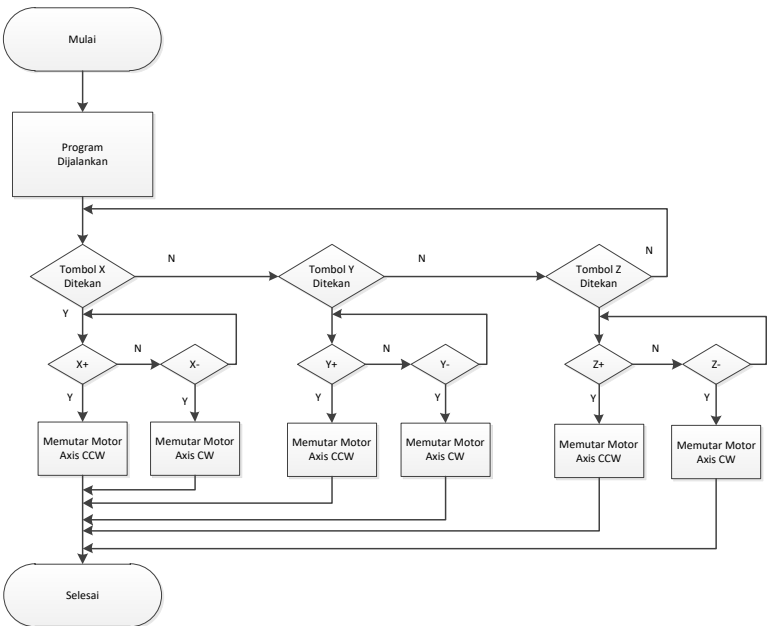
Bagian pemrograman ini digunakan untuk pengiriman data secara serial, dari komputer ke mikrokontroller. Pengiriman data memanfaatkan pin TX dan RX yang dimiliki arduino. Data yang dikirim biasanya merupakan kode numerik untuk penggerak mesin CNC. Untuk lebih jelasnya program dapat dilihat pada Lampiran A

3.3.2 Program *Interface* Menggunakan Visual Basic (VB)

Untuk bagian pemrograman *interface* sendiri terbagi menjadi dua bagian, tetapi masih dalam satu layar kerja, yang pertama bagian penggerak manual (pengaturan gerakan) dan yang kedua adalah koneksi antara *interface* dengan arduino serta penggerakan mesin CNC menggunakan G-code.

Seperti yang tampak pada Gambar 3.20 , motor stepper digerakkan secara manual dengan bergantian. Masing-masing dari motor stepper mewakili sumbu pergerakan dari CNC. Baik sumbu X, sumbu Y, maupun sumbu Z masing-masing memiliki dua gerakan yaitu CW (*Clock Wise*) dan CCW (*Counter Clock Wise*) berikut merupakan. Berikut merupakan

beberapa bagian program penyusun dari kerja sesuai dengan flow chart yang telah ditampilkan pada Gambar 3.19.



Gambar 3.19 Flow Chart Pergerakan Motor Stepper

```
If RB10.Checked Then
    TextBox2.Text = "G00 X" & (Mid(TextBox2.Text, 6, 7) + 10)
    MsgBox(TextBox2.Text)
    SerialPort1.WriteLine(TextBox2.Text)
```

Gambar 3.20 Pemrograman Tombol X+

• **Tombol X Plus**

Pada program yang tampak pada Gambar 3.20, digunakan tiga kemungkinan, masing-masing berdasarkan dengan seberapa jauh pergerakan yang kita inginkan. Yang pertama dengan menggunakan “RB10.Checked”, RB10 ini merupakan nama dari dialog yang digunakan sehingga apabila kotak dialog ini diaktifkan maka pergerakan akan bernilai +10. “RB5.checked” RB5 merupakan nama dari dialog yang digunakan sehingga apabila kotak ini diaktifkan maka pergerakan akan

bernilai +5. “RB1.Checked” RB1 juga nama dari dialog yang digunakan, sehingga apabila dialog ini aktif maka pergerakan akan bernilai +1.

Saat pemilihan dari besar pergerakan yang diinginkan, mesin CNC tidak langsung bergerak, melainkan menunggu tombol “textBox2.Text” ditekan maka nilai dari pergerakannya akan menjadi “=G00 X10” atau “=G00 X5” atau “=G00 X1” sesuai dengan pilihan yang dilakukan sebelumnya. G00 merupakan kode mesin CNC yang mengindikasikan untuk pergerakan motor stepper, sedangkan untuk X merupakan arah dari pergerakan mesin CNC. Untuk “MsgBox(textBox2.Text)” merupakan text yang akan keluar setelah ditekan tombol “textBox2.Text” sebagai nilai dari pergerakan yang akan dilakukan (pemberitahuan kepada pengguna). Kondisi ini berlaku untuk semua nilai pergerakan yang diinginkan.

- **Tombol Y Plus**

```
If RB10.Checked Then
    TextBox3.Text = "G00 Y" & (Mid(TextBox3.Text, 6, 7) + 10)
    MsgBox(TextBox3.Text)
    SerialPort1.WriteLine(TextBox3.Text)
ElseIf RB5.Checked Then
    TextBox3.Text = "G00 Y" & (Mid(TextBox3.Text, 6, 7) + 5)
    MsgBox(TextBox3.Text)
    SerialPort1.WriteLine(TextBox3.Text)
ElseIf RB1.Checked Then
    TextBox3.Text = "G00 Y" & (Mid(TextBox3.Text, 6, 7) + 1)
    MsgBox(TextBox3.Text)
    SerialPort1.WriteLine(TextBox3.Text)
```

Gambar 3.21 Pemrograman Tombol Y+

Pada program yang tampak pada Gambar 3.21, juga digunakan tiga kemungkinan, masing-masing berdasarkan dengan seberapa jauh pergerakan yang kita inginkan. Yang pertama dengan menggunakan “RB10.Checked”, RB10 ini merupakan nama dari dialog yang digunakan sehingga apabila kotak dialog ini diaktifkan maka pergerakan akan bernilai +10. “RB5.checked” RB5 merupakan nama dari dialog yang digunakan sehingga apabila kotak ini diaktifkan maka pergerakan akan bernilai +5. “RB1.Checked” RB1 juga nama dari dialog yang digunakan, sehingga apabila dialog ini aktif maka pergerakan akan bernilai +1.

Saat pemilihan dari besar pergerakan yang diinginkan, mesin CNC tidak langsung bergerak, melainkan menunggu tombol “textBox3.Text” ditekan maka nilai dari pergerakannya akan menjadi “=G00 Y10” atau

“=G00 Y5” atau “=G00 Y1” sesuai dengan pilihan yang dilakukan sebelumnya. G00 merupakan kode mesin CNC yang mengindikasikan untuk pergerakan motor stepper, sedangkan untuk Y merupakan arah dari pergerakan mesin CNC. Untuk “MsgBox(TextBox3.Text)” merupakan text yang akan keluar setelah ditekan tombol “TextBox3.Text” sebagai nilai dari pergerakan yang akan dilakukan (pemberitahuan kepada pengguna). Kondisi ini berlaku untuk semua nilai pergerakan yang diinginkan.

- **Tombol Z Plus**

```
If RB10.Checked Then
    TextBox4.Text = "G00 Z" & (Mid(TextBox4.Text, 6, 7) + 10)
    MsgBox(TextBox4.Text)
    SerialPort1.WriteLine(TextBox4.Text)
ElseIf RB5.Checked Then
    TextBox4.Text = "G00 Z" & (Mid(TextBox4.Text, 6, 7) + 5)
    MsgBox(TextBox4.Text)
    SerialPort1.WriteLine(TextBox2.Text)
ElseIf RB1.Checked Then
    TextBox4.Text = "G00 Z" & (Mid(TextBox4.Text, 6, 7) + 1)
    MsgBox(TextBox4.Text)
    SerialPort1.WriteLine(TextBox4.Text)
```

Gambar 3.22 Pemrograman Tombol Z+

Pada program yang tampak pada Gambar 3.22, juga digunakan tiga kemungkinan, masing-masing berdasarkan dengan seberapa jauh pergerakan yang kita inginkan. Yang pertama dengan menggunakan “RB10.Checked”, RB10 ini merupakan nama dari dialog yang digunakan sehingga apabila kotak dialog ini diaktifkan maka pergerakan akan bernilai +10.

“RB5.checked” RB5 meruakan nama dari dialog yang digunakan sehingga apabila kotak ini diaktifkan maka pergerakan akan bernilai +5. “RB1.Checked” RB1 juga nama dari dialog yang digunakan, sehingga apabila dialog ini aktif maka pergerakan akan bernilai +1.

Saat pemilihan dari besar pergerakan yang diinginkan, mesin CNC tidak langsung bergerak, melainkan menunggu tombol “textBox4.Text” ditekan maka nilai dari pergerakannya akan menjadi “=G00 Z10” atau “=G00 Z5” atau “=G00 Z1” sesuai dengan pilihan yang dilakukan sebelumnya. G00 merupakan kode mesin CNC yang mengindikasikan untuk pergerakan motor stepper, sedangkan untuk Z merupakan arah dari pergerakan mesin CNC. Untuk “MsgBox(TextBox4.Text)” merupakan

text yang akan keluar setelah ditekan tombol “TextBox4.Text” sebagai nilai dari pergerakan yang akan dilakukan (pemberitahuan kepada pengguna). Kondisi ini berlaku untuk semua nilai pergerakan yang diinginkan.

- **Tombol X Min**

```
If RB10.Checked Then
    TextBox2.Text = "G00 X" & (Mid(TextBox2.Text, 6, 7) - 10)
    MsgBox(TextBox2.Text)
    SerialPort1.WriteLine(TextBox2.Text)
ElseIf RB5.Checked Then
    TextBox2.Text = "G00 X" & (Mid(TextBox2.Text, 6, 7) - 5)
    MsgBox(TextBox2.Text)
    SerialPort1.WriteLine(TextBox2.Text)
ElseIf RB1.Checked Then
    TextBox2.Text = "G00 X" & (Mid(TextBox2.Text, 6, 7) - 1)
    MsgBox(TextBox2.Text)
    SerialPort1.WriteLine(TextBox2.Text)
```

Gambar 3.23 Pemrograman Tombol X-

Pada program yang tampak pada Gambar 3.23, juga digunakan tiga kemungkinan, masing-masing berdasarkan dengan seberapa jauh pergerakan yang kita inginkan. Yang pertama dengan menggunakan “RB10.Checked”, RB10

ini merupakan nama dari dialog yang digunakan sehingga apabila kotak dialog ini diaktifkan maka pergerakan akan bernilai +10. “RB5.checked” RB5 merupakan nama dari dialog yang digunakan sehingga apabila kotak ini diaktifkan maka pergerakan akan bernilai +5. “RB1.Checked” RB1 juga nama dari dialog yang digunakan, sehingga apabila dialog ini aktif maka pergerakan akan bernilai +1.

Saat pemilihan dari besar pergerakan yang diinginkan, mesin CNC tidak langsung bergerak, melainkan menunggu tombol “textBox2.Text” ditekan maka nilai dari pergerakannya akan menjadi “=G00 X-10” atau “=G00 X-5” atau “=G00 X-1” sesuai dengan pilihan yang dilakukan sebelumnya. G00 merupakan kode mesin CNC yang mengindikasikan untuk pergerakan motor stepper, sedangkan untuk X- merupakan arah dari pergerakan mesin CNC. Untuk “MsgBox(TextBox2.Text)” merupakan text yang akan keluar setelah ditekan tombol “TextBox2.Text” sebagai nilai dari pergerakan yang akan dilakukan (pemberitahuan kepada pengguna). Kondisi ini berlaku untuk semua nilai pergerakan yang diinginkan.

```

If RB10.Checked Then
    TextBox3.Text = "G00 Y" & (Mid(TextBox3.Text, 6, 7) - 10)
    MsgBox(TextBox3.Text)
    SerialPort1.WriteLine(TextBox3.Text)
ElseIf RB5.Checked Then
    TextBox3.Text = "G00 Y" & (Mid(TextBox3.Text, 6, 7) - 5)
    MsgBox(TextBox3.Text)
    SerialPort1.WriteLine(TextBox3.Text)
ElseIf RB1.Checked Then
    TextBox3.Text = "G00 Y" & (Mid(TextBox3.Text, 6, 7) - 1)
    MsgBox(TextBox3.Text)
    SerialPort1.WriteLine(TextBox3.Text)

```

Gambar 3.24 Pemrograman Tombol Y-

- **Tombol Y Min**

Pada program yang tampak pada Gambar 3.24, juga digunakan tiga kemungkinan, masing-masing berdasarkan dengan seberapa jauh pergerakan yang kita inginkan. Yang pertama dengan menggunakan “RB10.Checked”, RB10 ini merupakan nama dari dialog yang digunakan sehingga apabila kotak dialog ini diaktifkan maka pergerakan akan bernilai +10. “RB5.Checked” RB5 merupakan nama dari dialog yang digunakan sehingga apabila kotak ini diaktifkan maka pergerakan akan bernilai +5. “RB1.Checked” RB1 juga nama dari dialog yang digunakan, sehingga apabila dialog ini aktif maka pergerakan akan bernilai +1.

Saat pemilihan dari besar pergerakan yang diinginkan, mesin CNC tidak langsung bergerak, melainkan menunggu tombol “TextBox3.Text” ditekan maka nilai dari pergerakannya akan menjadi “=G00 Y-10” atau “=G00 Y-5” atau “=G00 Y-1” sesuai dengan pilihan yang dilakukan sebelumnya. G00 merupakan kode mesin CNC yang mengindikasikan untuk pergerakan motor stepper, sedangkan untuk Y- merupakan arah dari pergerakan mesin CNC. Untuk “MsgBox(TextBox3.Text)” merupakan text yang akan keluar setelah ditekan tombol “TextBox3.Text” sebagai nilai dari pergerakan yang akan dilakukan (pemberitahuan kepada pengguna). Kondisi ini berlaku untuk semua nilai pergerakan yang diinginkan.


```

If RB10.Checked Then
    TextBox4.Text = "G00 Z" & (Mid(TextBox4.Text, 6, 7) - 10)
    MsgBox(TextBox4.Text)
    SerialPort1.WriteLine(TextBox4.Text)
ElseIf RB5.Checked Then
    TextBox4.Text = "G00 Z" & (Mid(TextBox4.Text, 6, 7) - 5)
    MsgBox(TextBox4.Text)
    SerialPort1.WriteLine(TextBox4.Text)
ElseIf RB1.Checked Then
    TextBox4.Text = "G00 Z" & (Mid(TextBox4.Text, 6, 7) - 1)
    MsgBox(TextBox4.Text)
    SerialPort1.WriteLine(TextBox4.Text)

```

Gambar 3.25 Pemrograman Tombol Z-

- **Tombol Z Min**

Pada program yang tampak pada Gambar 3.25, digunakan tiga kemungkinan, masing-masing berdasarkan dengan seberapa jauh pergerakan yang kita inginkan. Yang pertama dengan menggunakan “RB10.Checked”, RB10 ini merupakan nama dari dialog yang digunakan sehingga apabila kotak dialog ini diaktifkan maka pergerakan akan bernilai +10. “RB5.Checked” RB5 merupakan nama dari dialog yang digunakan sehingga apabila kotak ini diaktifkan maka pergerakan akan bernilai +5. “RB1.Checked” RB1 juga nama dari dialog yang digunakan, sehingga apabila dialog ini aktif maka pergerakan akan bernilai +1.

Saat pemilihan dari besar pergerakan yang diinginkan, mesin CNC tidak langsung bergerak, melainkan menunggu tombol “textBox4.Text” ditekan maka nilai dari pergerakannya akan menjadi “=G00 Z-10” atau “=G00 Z-5” atau “=G00 Z-1” sesuai dengan pilihan yang dilakukan sebelumnya. G00 merupakan kode mesin CNC yang mengindikasikan untuk pergerakan motor stepper, sedangkan untuk Z- merupakan arah dari pergerakan mesin CNC. Untuk “MsgBox(TextBox4.Text)” merupakan text yang akan keluar setelah ditekan tombol “TextBox4.Text” sebagai nilai dari pergerakan yang akan dilakukan (pemberitahuan kepada pengguna). Kondisi ini berlaku untuk semua nilai pergerakan yang diinginkan.

- **Pengaktifan *Spindel***

```

If TextBox3.Text = "A" Then
    TextBox4.Text = "M03"
    TextBox3.Text = "B"
    SerialPort1.WriteLine(TextBox4.Text)
Else
    TextBox4.Text = "M05"
    TextBox3.Text = "A"
    SerialPort1.WriteLine(TextBox4.Text)

```

Gambar 3.26 Program Memulai dan Menghentikan Putaran *Spindel*

Pada program yang tampak pada Gambar 3.26 digunakan untuk mengaktifkan motor DC yang digunakan sebagai spindle (Bor). Program ini memiliki dua kemungkinan, yang pertama adalah nilai “A=M05” dan nilai dari “B=M03”. M05 merupakan salah satu kode numerik yang dapat dikenali oleh mesin CNC untuk menghentikan perputaran dari *spindel*.

Sedangkan untuk M03 juga merupakan salah satu kode numerik yang dikenali oleh mesin CNC untuk memulai putaran pada *spindel*. Pada awal program dijalankan nilai dari *spindel* bernilai A, setelah textBox ditekan maka nilai dari TextBox3.Text akan berubah menjadi B dan nilainya akan menjadi M03, dan nilai dari A tidak diaktifkan hingga tombol yang sama ditekan kembali sehingga nilainya akan berubah menjadi A.

- **Peletakan Kondisi Awal**

```

TextBox2.Text = "G00 X0"
TextBox3.Text = "G00 Y0"
TextBox4.Text = "G00 Z0"
TextBox5.Text = "G00 X-0"
TextBox6.Text = "G00 Y-0"
TextBox7.Text = "G00 Z-0"
SerialPort1.WriteLine("G00 X0 Y0 Z0")

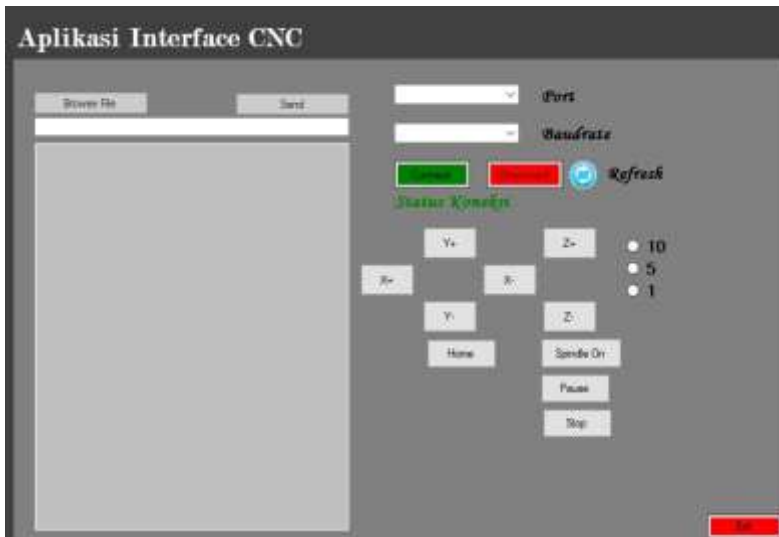
```

Gambar 3.27 Program Pengembalian ke Posisi Awal

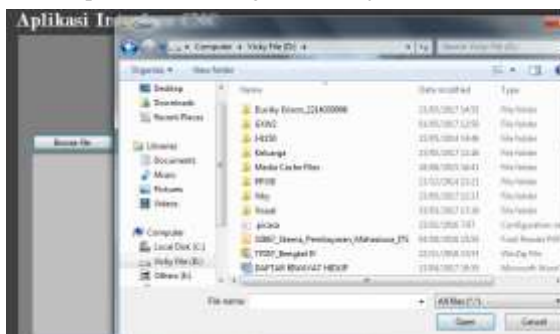
Program yang tampak pada Gambar 3.27 digunakan untuk mengembalikan kondisi awal dari mesin CNC setelah bergerak. Dengan mengaktifkan program ini maka semua nilai kondisi pergerakan akan diarahkan menjadi 0 (titik awal sebelum bergerak). “SerialPort1.Writeline” digunakan untuk mengirimkan perintah ke arduino sehingga akan diteruskan ke driver motor yang akan mengakibatkan pergerakan dari mesin CNC. “G00 X0 Y0 Z0” merupakan nilai yang dikirimkan, G00 merupakan kode numerik untuk menjalankan

motor, untuk X0 Y0 Z0 merupakan koordinat terakhir yang dituju dalam mesin CNC.

Pada Gambar 3.28 adalah tampilan awal dari program visual basic setelah dijalankan dimana pada tampilan tersebut terdapat beberapa menu untuk menggerakkan CNC serta pemilihan port dan hal-hal lain yang dibutuhkan untuk menjalankan mesin CNC.



Gambar 3.28 Tampilan Setelah Program VB Dijalankan



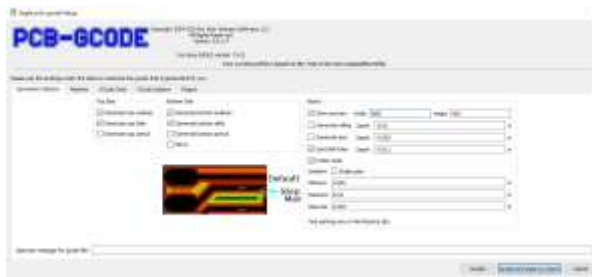
Gambar 3.29 Tampilan ketika Browse File Ditekan

2. Setelah format board telah selesai dibuat, tekan run ULP seperti yang tampak pada Gambar 3.31 Yang bertanda merah



Gambar 3.31 Icon Run ULP

3. Setelah ditekan maka akan mengeluarkan windows explor untuk mencari file library yang sebelumnya telah kita instal. Kemudian pilih file *pcb-G-code-setup.ulp* dengan demikian windows baru akan muncul seperti pada Gambar 3.32



Gambar 3.32 Tampilan Awal PCB-G-code

4. Centang bagian Top dan Bottom untuk mendapatkan hasil kode numerik pada bagian seluruh jalur dan bagian peletakan komponen.
5. Setelah itu atur kedalaman pengeboran dengan mengubah nilai dari Z. Seperti yang tampak pada Gambar 3.33 Yang bertanda merah.



Gambar 3.33 Pemberian Nilai Pada Sumbu Z

6. Kemudian klik *Accept* untuk mengakhiri proses perubahan yang telah dilakukan. File yang telah dirubah akan tersimpan pada folder yang sama dengan penyimpanan skematik yang telah dirubah menjadi kode numerik.

3.4 Hasil Pembuatan

Setelah melakukan sebuah perancangan mesin CNC, hasil dari perancangan dapat dilihat pada Gambar 3.30 dan Gambar 3.31

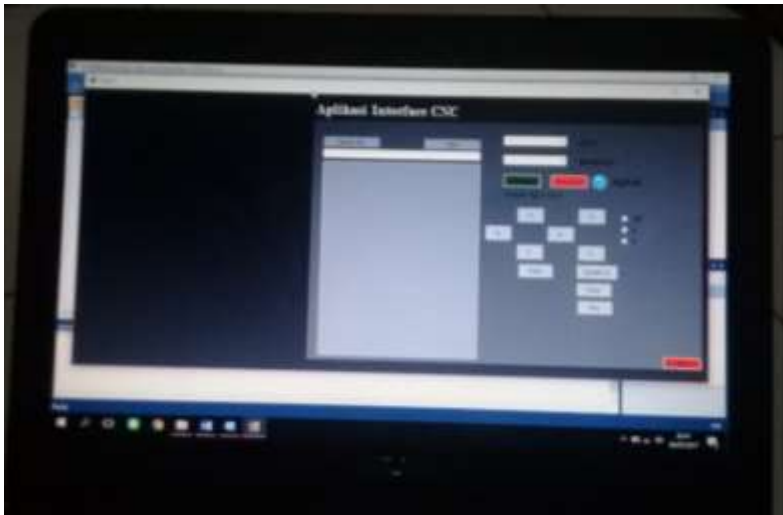


Gambar 3.34 Mesin CNC Tampak Depan



Gambar 3.35 Mesin CNC Tampak Samping

Untuk Hasil pembuatan Interface yang menghubungkan antara Laptop dengan CNC dapat dilihat pada Gambar 3.32



Gambar 3.36 Tampilan Aplikasi *Interface* CNC

-----Halaman ini sengaja dikosongkan-----

BAB IV

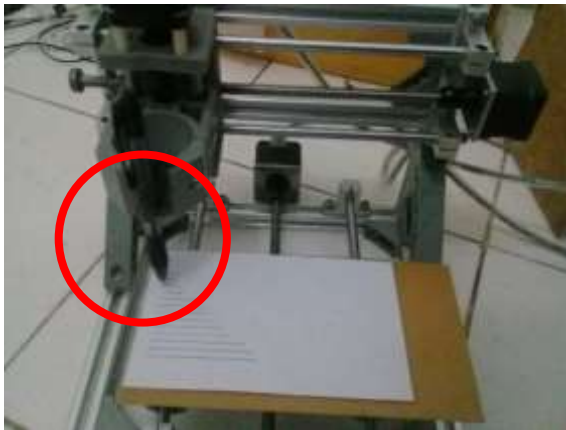
HASIL DAN IMPLEMENTASI

Pengujian pada mesin CNC ini terdiri dari beberapa bagian yang juga mendukung kerja dari mesin CNC, diantaranya adalah pengujian akurasi perpindahan stepper, pengujian perpindahan motor stepper menggunakan tatap muka visual basic, serta pengujian keseluruhan. Berikut akan dibahas mengenai masing-masing pengujian.

4.1 Pengujian Akurasi

Pengujian ini dilakukan berdasarkan perhitungan dengan menentukan besar perpindahan sesuai dengan banyak step yang dilakukan. Yang kemudian akan dibandingkan dengan pergerakan pada motor stepper setelah digerakkan menggunakan program dari arduino. Dari perbandingan tersebut akan didapat selisih nilai antara hasil dari perpindahan yang terukur dan hasil perpindahan dengan menggunakan perhitungan secara manual.

Pengujian ini dilakukan dengan cara memberikan bolpen sebagai penanda yang ditaruh pada tempat spindel, seperti yang tampak pada Gambar 4.1. Kemudian masing-masing dari motor stepper akan diberikan step secara bertahap, mulai dari 20 step hingga 700 step. Kemudian pengukuran dilakukan dengan penggaris mulai dari titik awal tanda bolpen, hingga titik akhir setelah motor stepper diberikan step.



Gambar 4.1 Penempatan Bolpen Pada Mesin CNC

Motor stepper yang digunakan dalam mesin CNC ini merupakan jenis NEMA 17. Yang memiliki 200 step dalam satu putaran sehingga untuk setiap step dalam motor ini besarnya $1,8^{\circ}$. Untuk menggerakkan *end effector* dari ketiga sumbu pada mesin CNC ini menggunakan *balscrew* dengan besar diameter 6 mm. Sehingga dengan menggunakan persamaan keliling lingkaran maka besar dari keliling lingkaran adalah 18,84 mm.

Dengan demikian dalam setiap step dari perpindahan dalam motor stepper adalah :

$$\begin{aligned}
 &= \frac{\text{Banyak Step}}{200} \times \text{Keliling lingkaran} \\
 &= \frac{1}{200} \times 18,84 \\
 &= 0,0942 \text{ mm}
 \end{aligned}$$

Sehingga apabila ingin melakukan perpindahan sejauh 1 mm maka banyak step yang dibutuhkan adalah

$$= \frac{1}{0,0942} = 10,6 \text{ step}$$

Berdasarkan hal tersebut berikut merupakan pengukuran yang telah didapat dari pergerakan stepper, kemudian dibandingkan dengan hasil perhitungannya. Percobaan ini akan dilakukan pada sumbu X, sumbu Y, dan sumbu Z.

4.1.1 Sumbu X

Sumbu X akan berjalan sesuai dengan banyak step yang telah diberikan kemudian akan dihitung besar perpindahannya mulai titik awal hingga berakhirnya garis dari bolpen sebagai media penanda.



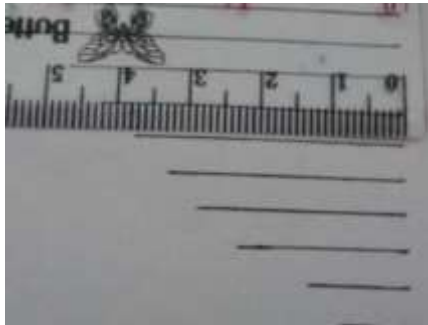
Gambar 4.2 Panjang Sumbu X Setelah 20 Step

Pada Gambar 4.2 dapat dilihat setelah pengukuran data pertama diberikan 20 step pada motor stepper, maka perpindahan yang dialami oleh *end effector* yang terukur sejauh 1,9 mm sedangkan menurut perhitungan bernilai 1,88 mm.



Gambar 4.3 Panjang Sumbu X Setelah 140 Step

Pada Gambar 4.3 dapat dilihat setelah pengukuran data pertama diberikan 140 step pada motor stepper, maka perpindahan yang dialami oleh *end effector* yang terukur sejauh 13,3 mm sedangkan menurut perhitungan bernilai 13,19 mm.



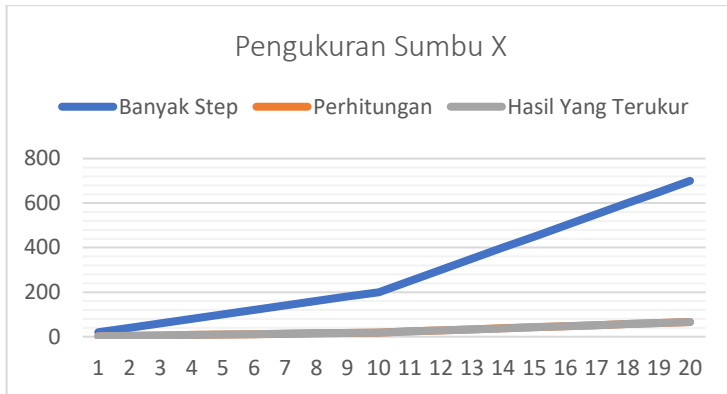
Gambar 4.4 Panjang Sumbu X Setelah 400 Step

Pada Gambar 4.4 dapat dilihat setelah pengukuran data pertama diberikan 400 step pada motor stepper, maka perpindahan yang dialami oleh *end effector* yang terukur sejauh 37,5 mm sedangkan menurut perhitungan bernilai 37,68 mm. Untuk keseluruhan data dapat dilihat pada Tabel 4.1

Tabel 4.1 Data Hasil Pengujian Motor Stepper X

Banyak Step	Hasil Perhitungan (mm)	Hasil Pengukuran (mm)	Error (mm)
20	1,88	1,9	0,02
40	3,77	3,6	0,17
60	5,65	5,6	0,05
80	7,54	7,7	0,16
100	9,42	9,5	0,08
120	11,30	11,3	0,00
140	13,19	13,3	0,11
160	15,07	15	0,07
180	16,96	17	0,04
200	18,84	18,7	0,14
250	23,55	23,6	0,05
300	28,26	28,1	0,16
350	32,97	33	0,03
400	37,68	37,5	0,18
450	42,39	42,4	0,01
500	47,10	46,9	0,20
550	51,81	51,5	0,31
600	56,52	56,5	0,02
650	61,23	61,4	0,17
700	65,94	65,8	0,14

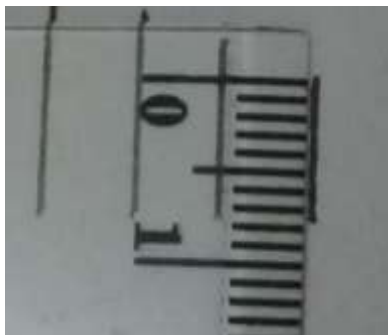
Terjadinya error pada data yang tampak pada Tabel 4.1 kemungkinan terjadi karena ketelitian dalam pengukurannya. Karena pengukurannya dilakukan dengan menggunakan penggaris mistar yang memiliki ketelitian 1 mm. Dari rata-rata data yang didapat hampir keseluruhan mendekati dengan nilai yang didapat dengan perhitungan. Pada Gambar 4.5 dapat dilihat kurva yang mewakili dari keseluruhan data yang didapat pada percobaan sumbu X.



Gambar 4.5 Perbandingan Dari Seluruh Data Pada Sumbu X

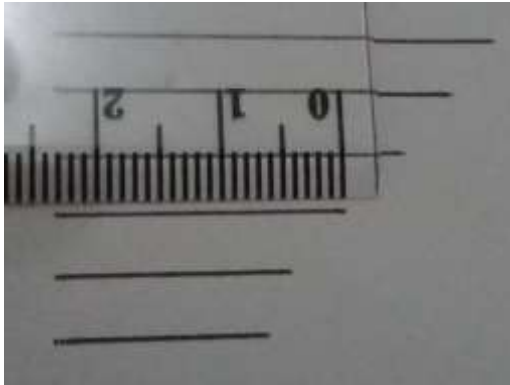
4.1.2 Sumbu Y

Sumbu Y akan berjalan sesuai dengan banyak step yang telah diberikan kemudian akan dihitung besar perpindahannya mulai titik awal hingga berakhirnya garis yang didapat dari bolpen sebagai media penanda.



Gambar 4.6 Panjang Sumbu Y Setelah 80 Step

Pada Gambar 4.6 dapat dilihat setelah pengukuran data pertama diberikan 80 step pada motor stepper, maka perpindahan yang dialami oleh *end effector* yang terukur sejauh 7,8 mm sedangkan menurut perhitungan bernilai 7,54 mm.



Gambar 4.7 Panjang Sumbu Y Setelah 250 Step

Pada Gambar 4.7 dapat dilihat setelah pengukuran data pertama diberikan 250 step pada motor stepper, maka perpindahan yang dialami oleh *end effector* yang terukur sejauh 23,4 mm sedangkan menurut perhitungan bernilai 23,55 mm.



Gambar 4.8 Panjang Sumbu Y Setelah 350 Step

Pada Gambar 4.8 dapat dilihat setelah pengukuran data pertama diberikan 350 step pada motor stepper, maka perpindahan yang dialami oleh *end effector* yang terukur sejauh 33 mm sedangkan menurut perhitungan bernilai 32,97 mm. Untuk keseluruhan data dapat dilihat pada Tabel 4.2

Tabel 4.2 Data Hasil Pengujian Motor Stepper Y

Banyak Step	Hasil Perhitungan (mm)	Hasil Pengukuran (mm)	Error (mm)
20	1,88	1,9	0,02
40	3,77	3,7	0,07
60	5,65	5,5	0,15
80	7,54	7,8	0,26
100	9,42	9,6	0,18
120	11,30	11,5	0,20
140	13,19	13	0,19
160	15,07	15,2	0,13
180	16,96	16,9	0,06
200	18,84	18,6	0,24
250	23,55	23,4	0,15
300	28,26	28,4	0,14
350	32,97	33	0,03
400	37,68	37,5	0,18
450	42,39	42,5	0,11
500	47,10	47,3	0,20
550	51,81	51,9	0,09
600	56,52	56,4	0,12
650	61,23	61,4	0,17
700	65,94	66	0,06

Terjadinya error pada data yang tampak pada Tabel 4.2 kemungkinan terjadi karena ketelitian dalam pengukurannya. Karena pengukurannya dilakukan dengan menggunakan penggaris mistar yang memiliki ketelitian 1 mm. Dari rata-rata data yang didapat hampir keseluruhan mendekati dengan nilai yang didapat dengan perhitungan, walaupun terdapat perbedaan, tetapi perbedaan tersebut selisihnya masih kurang dari 0,3 mm. Untuk lebih jelasnya dapat dilihat pada Gambar 4.9 yang menampilkan seluruh data yang telah didapat dengan bentuk kurva.



Gambar 4.9 Perbandingan Dari Seluruh Data Pada Sumbu Y

4.1.3 Sumbu Z

Sumbu Z akan berjalan sesuai dengan banyak step yang telah diberikan kemudian akan dihitung besar perpindahannya mulai titik awal hingga berakhirnya garis dari bolpen sebagai media penanda.



Gambar 4.10 Panjang Sumbu Z Setelah 120 Step

Pada Gambar 4.10 dapat dilihat setelah pengukuran data diberikan 120 step pada motor stepper, maka perpindahan yang dialami oleh *end effector* yang terukur sejauh 11,5 mm sedangkan menurut perhitungan bernilai 11,30 mm.



Gambar 4.11 Panjang Sumbu Z Setelah 180 Step

Pada Gambar 4.11 dapat dilihat setelah pengukuran data diberikan 180 step pada motor stepper, maka perpindahan yang dialami oleh *end effector* yang terukur sejauh 16,8 mm sedangkan menurut perhitungan bernilai 16,96 mm.



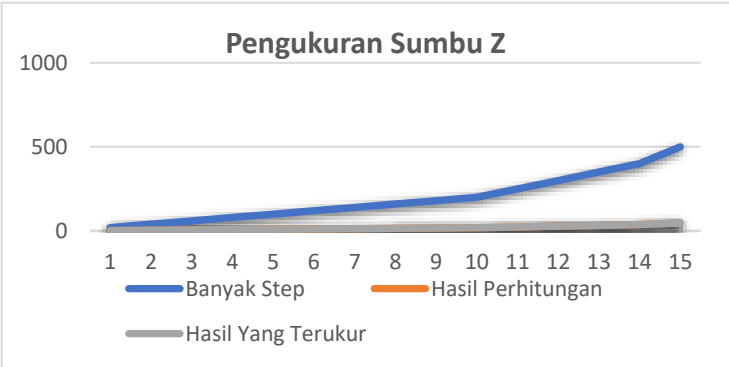
Gambar 4.12 Panjang Sumbu Z Setelah 500 Step

Pada Gambar 4.12 dapat dilihat setelah pengukuran data diberikan 500 step pada motor stepper, maka perpindahan yang dialami oleh *end effector* yang terukur sejauh 47,2 mm sedangkan menurut perhitungan bernilai 47,10 mm. Untuk keseluruhan data dapat dilihat pada Tabel 4.3

Tabel 4.3 Data Hasil Pengujian Motor Stepper Z

Banyak Step	Hasil Perhitungan (mm)	Hasil Pengukuran (mm)	Error (mm)
20	1,88	1,8	0,08
40	3,77	3,5	0,27
60	5,65	5,8	0,15
80	7,54	7,3	0,24
100	9,42	9,6	0,18
120	11,30	11,5	0,20
140	13,19	13,2	0,01
160	15,07	15,2	0,13
180	16,96	16,8	0,16
200	18,84	18,9	0,06
250	23,55	23,7	0,15
300	28,26	28,4	0,14
350	32,97	32,8	0,17
400	37,68	37,5	0,18
500	47,10	47,2	0,10

Terjadinya error pada data yang tampak pada Tabel 4.3 kemungkinan terjadi karena ketelitian dalam pengukurannya. Karena pengukurannya dilakukan dengan menggunakan penggaris mistar yang memiliki ketelitian 1 mm. Dari rata-rata data yang didapat hampir keseluruhan mendekati dengan nilai yang didapat dengan perhitungan, walaupun terdapat perbedaan, tetapi perbedaan tersebut selisihnya masih kurang dari 0,3 mm. Untuk lebih jelasnya dapat dilihat pada Gambar 4.13 yang menampilkan seluruh data yang telah didapat dengan bentuk kurva.



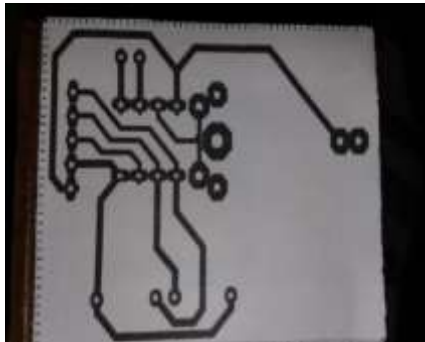
Gambar 4.13 Perbandingan Dari Seluruh Data Pada Sumbu Z

4.2 Pengujian Keseluruhan

Pengujian secara keseluruhan ini dilakukan dengan melakukan pengeboran pada PCB yang telah diberikan sirkuit elektronik, terdiri dari 3 rangkaian yang berbeda yaitu rangkaian RTC, rangkaian *driver relay*, dan rangkaian *amplifire*.

4.2.1 Pengujian Pengeboran Pada Rangkaian RTC

Pengujian ini dilakukan dengan cara memasukkan *G-code* kedalam tatapmuka pada visual basic, sehingga mesin CNC akan melakukan pergerakan dan pengeboran secara otomatis. Selain itu juga dipersiapkan untuk sebuah rangkaian RTC sebagai media yang akan dibor untuk tempat komponennya. Rangkaian RTC yang digunakan telah dilakukan mirror saat dikonfert menjadi pdf, kemudian diprint dan ditempelkan pada PCB yang telah disiapkan. Pengambilan data yang dilakukan adalah dengan menghitung banyak lubang yang telah terbor sesuai dengan rangkaian RTC yang telah disediakan.



Gambar 4.14 Rangkaian RTC Sebelum Dibor



Gambar 4.15 Rangkaian RTC Saat Dilakukan Proses Pengeboran



Gambar 4.16 Rangkaian RTC Tampak Atas Setelah Dibor

Pada Gambar 4.14 Merupakan rangkaian RTC, setelah dilakukan proses pengeboran maka hasil dari pengeboran tampak pada Gambar 4.15 dan Gambar 4.16, dapat dilihat banyak lubang yang ada pada rangkaian RTC sebanyak 26, telah berhasil dibor sebanyak 26 dan untuk ke-26 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan.

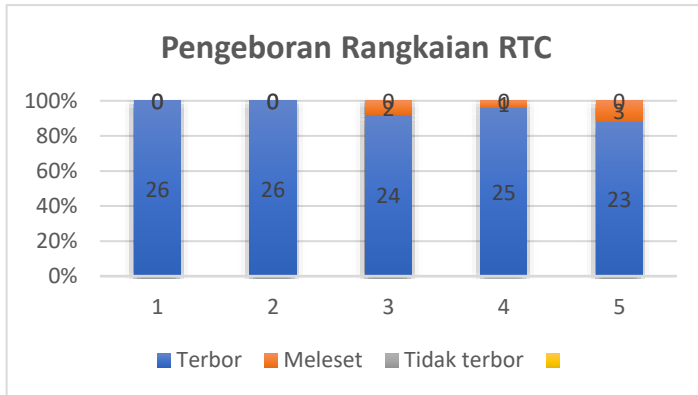
Setelah dilakukan percobaan pengeboran sebanyak lima kali berikut merupakan daftar yang telah didapat dalam percobaan yang telah dilakukan.

Tabel 4.4 Data Pengeboran Rangkaian RTC

Percobaan Ke-	Banyak Lubang	Keterangan			Ketepatan (%)
		Terbor	Meleset	Tidak terbor	
1	26	26	-	-	100
2	26	26	-	-	100
3	26	26	2	-	92,31
4	26	26	1	-	96,15
5	26	26	3	-	88,96

Dari data yang tampak pada Tabel 4.4 Menunjukkan bahwa ketepatan pengeboran paling banyak memiliki kesalahan sebanyak 11,04 % pada percobaan kelima sedangkan untuk sisanya kesalahan sudah kurang dari 10 %, bahkan ada tiga percobaan yang lubangnya sudah tepat sesuai dengan titik yang ditentukan. Dapat disimpulkan kemungkinan kesalahan tersebut terjadi karena ketepatan penempatan PCB saat akan dilakukan proses pengeboran, hal tersebut bisa terjadi karena apabila salah peletakan maka koordianat dari titik pengeboran yang telah ditentukan akan berpindah. Dengan demikian lubang yang akan terbor

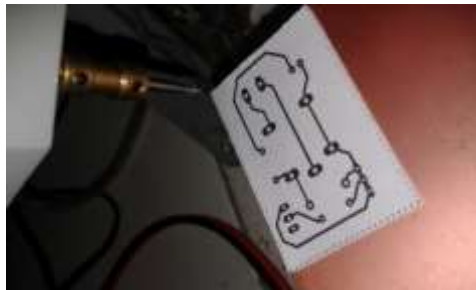
tidak sesuai dengan titik pengeboran yang seharusnya dilubangi. Untuk lebih jelasnya dapat dilihat pada Gambar 4.17 yang menjelaskan data yang telah diperoleh dari percobaan.



Gambar 4.17 Grafik Data Hasil Percobaan Rangkaian RTC

4.2.2 Pengujian Pengeboran Pada Rangkaian Driver Relay

Sama seperti rangkaian RTC diatas hasil dari routing Driver Relay di eagle dijadikan *G-code* dengan PCB *G-code*. Setelah itu rangkaian Driver Relay siap untuk di Bor.



Gambar 4.18 Rangkaian Driver Relay Tampak Atas Sebelum Dibor



Gambar 4.19 Rangkaian Driver Relay Tampak Atas Setelah Dibor



Gambar 4.20 Rangkaian Driver Relay Tampak Bawah Setelah Dibor

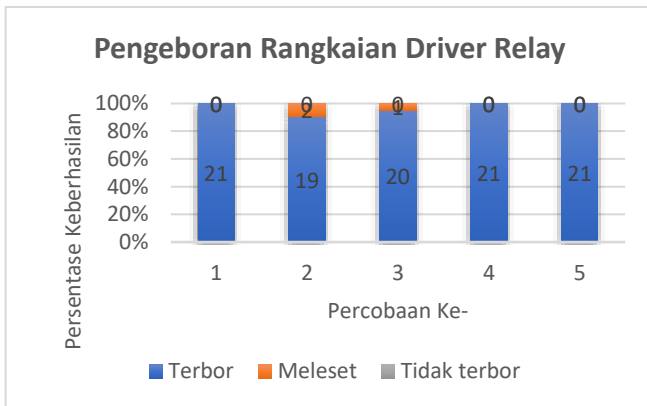
Pada Gambar 4.18 merupakan rangkaian driver relay setelah dilakukan pengeboran maka akan tampak seperti pada Gambar 4.19 dan Gambar 4.20, dapat dilihat banyak lubang yang ada pada rangkaian Driver Relay sebanyak 21, telah berhasil dibor sebanyak 21 dan untuk 21 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan.

Setelah dilakukan percobaan pengeboran sebanyak lima kali berikut merupakan daftar data yang telah didapat dalam percobaan yang telah dilakukan.

Tabel 4.5 Data Pengeboran Rangkaian Driver Relay

Percobaan Ke-	Banyak Lubang	Keterangan			Ketepatan (%)
		Terbor	Meleset	Tidak terbor	
1	21	21	-	-	100
2	21	19	2	-	90,476
3	21	20	1	-	95,238
4	21	21	-	-	100
5	21	21	-	-	100

Dari data yang tampak pada Tabel 4.5 Menunjukkan bahwa ketepatan pengeboran paling banyak memiliki kesalahan diatas sekitar 10%. Dapat disimpulkan kemungkinan kesalahan tersebut terjadi karena ketepatan penempatan PCB saat akan dilakukan proses pengeboran, hal tersebut bisa terjadi karena apabila salah peletakan maka koordianat dari titik pengeboran yang telah ditentukan akan berpindah. Dengan demikian lubang yang akan terborpun tidak sesuai dengan titik pengeboran yang seharusnya dilubangi. Untuk lebih jelasnya dapat dilihat pada Gambar 4.21.



Gambar 4.21 Grafik Data Hasil Percobaan Rangkaian RTC

4.2.3 Pengujian Pengeboran Pada Rangkaian Amplifier

Sama seperti rangkaian sebelumnya hasil dari routing Driver Amplifier di eagle ini dijadikan *G-code* dengan PCB *G-code*. Setelah itu rangkaian Amplifier siap untuk di Bor.



Gambar 4.22 Rangkaian Amplifier Tampak Atas Sebelum Dibor



Gambar 4.23 Rangkaian Amplifier Tampak Atas Setelah Dibor



Gambar 4.24 Rangkaian Amplifier Tampak Bawah Setelah Dibor

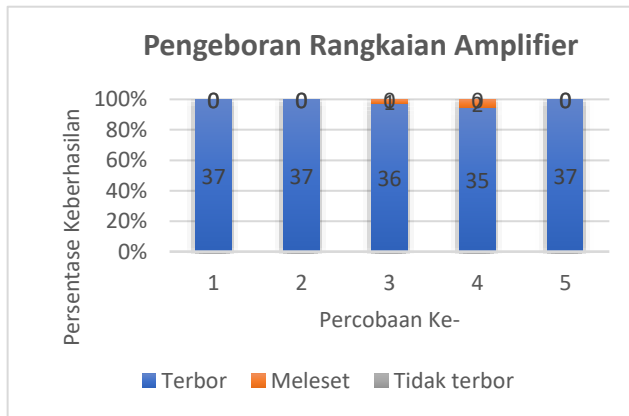
Pada Gambar 4.22 merupakan rangkaian dari *amplifier* setelah dilakukan proses pengeboran maka akan tampak seperti pada Gambar 4.23 dan Gambar 4.24, dapat dilihat banyak lubang yang ada pada rangkaian Amplifier sebanyak 37, telah berhasil dibor sebanyak 37 dan untuk 37 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan.

Setelah dilakukan percobaan pengeboran sebanyak lima kali, berikut merupakan daftar data yang telah didapat dalam percobaan yang telah dilakukan.

Tabel 4.6 Data Pengeboran Rangkaian Amplifier

Percobaan Ke-	Banyak Lubang	Keterangan			Ketepatan (%)
		Terbor	Meleset	Tidak terbor	
1	37	37	-	-	100
2	37	37	-	-	100
3	37	36	1	-	97,297
4	37	35	2	-	94,595
5	37	37	-	-	100

Dari data yang tampak pada Tabel 4.6 Menunjukkan bahwa ketepatan pengeboran paling banyak memiliki kesalahan sekitar 10%. Dapat disimpulkan kemungkinan kesalahan tersebut terjadi karena ketepatan penempatan PCB saat akan dilakukan proses pengeboran, hal tersebut bisa terjadi karena apabila salah peletakan maka koordianat dari titik pengeboran yang telah ditentukan akan berpindah. Dengan demikian lubang yang akan terborpun tidak sesuai dengan titik pengeboran yang seharusnya dilubangi. Berikut merupakan rangkuman dari keseluruhan data pada Gambar 4.25 yang telah didapat dalam hasil percobaan.



Gambar 4.25 Grafik Data Hasil Percobaan Rangkaian RTC

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

5.1 Kesimpulan

Setelah dilakukan pengambilan data, berikut merupakan beberapa kesimpulan yang telah didapat:

1. Dari pengujian perpindahan motor stepper dengan menggunakan banyak step yang diberikan, didapatkan bahwa antara perhitungan dan pengukuran dari jarak perpindahan yang ditempuh. Untuk motor stepper yang berada pada sumbu X selisih antara perhitungan dan pengukuran memiliki perbedaan nilai antara 0,00 mm – 0,18 mm. Untuk motor stepper yang ada pada sumbu Y selisih antara perhitungan dan pengukuran memiliki perbedaan antara 0,02 mm – 0,24 mm. Sedangkan motor stepper pada sumbu Z selisih antara perhitungan dan pengukuran memiliki perbedaan nilai antara 0,01 mm – 0,28 mm.
2. Pada pengujian secara keseluruhan dari ketiga rangkaian yang disediakan beberapa masih terdapat kesalahan dalam pengeboran. itu disebabkan saat proses peletakan PCB pada tempat pengeboran tidak sesuai dengan koordinat (0, 0, 0) yang ada pada mesin CNC.

5.2 Saran

Saran untuk penelitian selanjutnya adalah lebih banyak dilakukan penelitian pada bagian mekanik, sehingga pergerakan mesin CNC lebih bagus dan halus sehingga penyimpangan menjadi kurang dari 5%. Karena pada alat ini diperlukan tingkat kepresisian yang cukup tinggi. Selain itu untuk perubahan bentuk gambar menjadi kode numerik masih menggunakan konverter yang ada pada eagle, mungkin dapat dikembangkan untuk mendapatkan kode numerik dari suatu rangkaian dengan menggunakan pengolahan citra digital.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] ____, GRBL. <https://github.com/grbl/grbl/wiki> (diakses pada tanggal 26 Februari 2017)
- [2] ____, Arduino, <https://www.arduino.cc> (Diakses tanggal 19 Maret 2017)
- [3] Elektronika Dasar, Prinsip Kerja Motor DC, <https://elektronika-dasar.web.id/prinsip-kerja-motor-dc/> (diakses pada tanggal 19 Maret 2017)
- [4] Hafidhillisan, Rizal, 2016, PERANCANGAN OPERATION PANEL SISTEM CNC (COMPUTER NUMERICAL CONTROL) PORTABLE PADA MESIN PLOTTER BERBASIS MIKROKONTROLER, Tugas Akhir Teknik Elektro, Institus Teknologi Sepuluh Nopember, Surabaya.
- [5] Kristanto, Andri, 2004, Rekayasa Perangkat Lunak (Konsep Dasar), Gava Media, Yogyakarta.
- [6] Rahdiyanta, Dwi, Modul CNC-4, Fakultas Teknik Universitas Negeri Yogyakarta
- [7] Rochim, Taufik, 1993, Teori dan Teknologi Proses Pemesinan, Laboratorium Teknik Produksi, Jurusan Mesin, FTI-ITB, Bandung.
- [8] Pamungkas, Tria.2010, Motor Sepper,<https://pamungkas99.wordpress.com/2010/03/06/motor-stepper> (diakses pada tanggal 19 Maret 2017)
- [9] Partner 3D, Motor Stepper; Pengertian, Cara Kerja dan Jenis-jenisnya, www.partner3d.com/motor-stepper-pengertian-cara-kerja-dan-jenis-jenisnya/ (diakses pada tanggal 19 Maret 2017)
- [10] Permana, Budi. 2010. Dasar Dasar Pemrograman Visual Basic 2010. <http://ilmukomputer.org/2012/12/12/dasar-dasar-pemrograman-visual-studio-2010> (diakses pada tanggal 3 Maret 2017)
- [11] Wahana Komputer, 2010, Tutorial 5 Hari Belajar Pemrograman Visual Basic 2010, Andi,Yogyakarta.
- [12] Widarto. 2008. Teknik Pemesinan untuk Sekolah Menengah Kejuruan, Departmen Pendidikan Nasional, Direktorat Pembinaan SMK, Jakarta.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN

A.1 Program Tatap Muka Visual Basic 2010

Option Explicit On

Option Infer Off

Imports System.Linq

Imports System

Imports System.IO

Imports System.ComponentModel

Imports System.Threading

Imports System.IO.Ports

Imports System.Math

Public Class Form1

 Inherits Form

 Dim XX() As String

 Dim A As String

 Dim B As String

 Dim X As String

 Dim Y As String

 Dim X1 As Integer

 Dim Y1 As Integer

 Dim X2 As Integer

 Dim Y2 As Integer

 Dim X3 As Integer

 Dim Y3 As Integer

 Dim G As Graphics = Me.CreateGraphics

 Dim myPort As Array

 Dim i As Integer

 Dim FD As New OpenFileDialog()

 Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

 Button3.Enabled = False

 Button2.Enabled = False

 Button1.Enabled = False

End Sub

"Tombol Connect

```

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    SerialPort1.PortName = ComboBox1.Text
    SerialPort1.BaudRate = ComboBox2.Text
    SerialPort1.Parity = IO.Ports.Parity.None
    SerialPort1.StopBits = IO.Ports.StopBits.One
    SerialPort1.DataBits = 8
    SerialPort1.Open()
    Label1.Text = "Connected"
    Button4.Enabled = False
    Button3.Enabled = True
    Button2.Enabled = True
    Button1.Enabled = True
End Sub

```

"Tombol Disconnect

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    SerialPort1.Close()
    Button4.Enabled = True
    Button3.Enabled = False
    Button2.Enabled = False
    Button1.Enabled = False
    Label1.Text = "Disconnected."
End Sub

```

"Tombol Open

```

Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
    Dim FD As New OpenFileDialog()
    FD.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
    If FD.ShowDialog = Windows.Forms.DialogResult.OK Then
        ListBox1.Items.Clear()
        ListBox1.Items.AddRange(IO.File.ReadAllLines(FD.FileName))
    End If
End Sub

Private Function KondisiPerintahSesuai(ByVal w As String) As
Boolean
    Return w.StartsWith("G00") OrElse w.StartsWith("G01")

```


End Function

"Tombol Send

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Refresh()

TextBox12.Text = String.Join(" ", ListBox1.Items.Cast(Of String).Where(Function(w) w.StartsWith("G00 ") OrElse w.StartsWith("G01 ") OrElse w.StartsWith("M03") OrElse w.StartsWith("M05") OrElse w.StartsWith("M02") OrElse w.StartsWith("G20") OrElse w.StartsWith("(") OrElse w.StartsWith("G21") OrElse w.StartsWith("G90") OrElse w.StartsWith("G91"))).ToArray())

pnlBoard.CreateGraphics().Clear(pnlBoard.BackColor)

TmrSend.Enabled = True

If tmrProses.Enabled Then tmrProses.Enabled = False

If btnProcess.Text = "Proses" Then

btnProcess.Text = "Stop"

nomorPerintah = 0

If (daftarPerintah.Count > 0) Then

daftarPerintah.Clear()

posisiPerintah.Clear()

pathPerintah.Clear()

ListBox1.Items.Clear()

pnlBoard.CreateGraphics().Clear(pnlBoard.BackColor)

End If

Dim lines() As String = IO.File.ReadAllLines((FD.FileName))

'ubah sesuai alamat file ulp

ListBox1.Items.AddRange(lines)

daftarPerintah = lines.Where(Function(w)

KondisiPerintahSesuai(w)).ToList()

posisiPerintah = lines.Select(Function(w, i)

If(KondisiPerintahSesuai(w, i, -1)).Where(Function(i) i > -1).ToList()

pathPerintah = daftarPerintah.Select(Function(s)

LokasiTapDariDaftarPerintah(s)).ToList()

```

        tmrProses.Enabled = True
    Else
        btnProcess.Text = "Proses"

    End If
End Sub

```

"Timer Pengirima Program ke Arduino

Private Sub TmrSen_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TmrSend.Tick

```

    On Error Resume Next
    If TextBox13.Text = "M02" Then
        SerialPort1.WriteLine(TextBox13.Text)
        TmrSend.Enabled = False
        TmrSend.Stop()
    Else
        TextBox14.Text = Val(TextBox14.Text) + 1
        B = Split(TextBox12.Text, "*")(TextBox14.Text)
        TextBox13.Text = B
        SerialPort1.WriteLine(TextBox13.Text)
    End If
End Sub

```

"Tombol X Plus

Private Sub BtnXPlus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnXPlus.Click

```

    If RB10.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) + 10)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    ElseIf RB5.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) + 5)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    ElseIf RB1.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) + 1)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    End If

```

End Sub

"Tombol Y Plus

Private Sub BtnYPlus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnYPlus.Click

 If RB10.Checked Then

 TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) + 10)

 MsgBox(TextBox10.Text)

 SerialPort1.WriteLine(TextBox10.Text)

 ElseIf RB5.Checked Then

 TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) + 5)

 MsgBox(TextBox10.Text)

 SerialPort1.WriteLine(TextBox10.Text)

 ElseIf RB1.Checked Then

 TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) + 1)

 MsgBox(TextBox10.Text)

 SerialPort1.WriteLine(TextBox10.Text)

 End If

End Sub

"Tombol Z Plus

Private Sub BtnZPlus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnZPlus.Click

 If RB10.Checked Then

 TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) + 10)

 MsgBox(TextBox9.Text)

 SerialPort1.WriteLine(TextBox9.Text)

 ElseIf RB5.Checked Then

 TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) + 5)

 MsgBox(TextBox9.Text)

 SerialPort1.WriteLine(TextBox9.Text)

 ElseIf RB1.Checked Then

 TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) + 1)

 MsgBox(TextBox9.Text)

 SerialPort1.WriteLine(TextBox9.Text)

 End If

End Sub

"Tombol X min

```

Private Sub BtnXMin_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnXMin.Click
    If RB10.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) - 10)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    ElseIf RB5.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) - 5)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    ElseIf RB1.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) - 1)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    End If
End Sub

```

"Tombol Y min

```

Private Sub BtnYMin_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnYMin.Click
    If RB10.Checked Then
        TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) - 10)
        MsgBox(TextBox10.Text)
        SerialPort1.WriteLine(TextBox10.Text)
    ElseIf RB5.Checked Then
        TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) - 5)
        MsgBox(TextBox10.Text)
        SerialPort1.WriteLine(TextBox10.Text)
    ElseIf RB1.Checked Then
        TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) - 1)
        MsgBox(TextBox10.Text)
        SerialPort1.WriteLine(TextBox10.Text)
    End If
End Sub

```

"Tombol Z min

```

Private Sub BtnZMin_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnZMin.Click
    If RB10.Checked Then

```

```

        TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) - 10)
        MsgBox(TextBox9.Text)
        SerialPort1.WriteLine(TextBox9.Text)
    ElseIf RB5.Checked Then
        TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) - 5)
        MsgBox(TextBox9.Text)
        SerialPort1.WriteLine(TextBox9.Text)
    ElseIf RB1.Checked Then
        TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) - 1)
        MsgBox(TextBox9.Text)
        SerialPort1.WriteLine(TextBox9.Text)
    End If
End Sub

'Spindle
Private Sub BtnSpn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnSpn.Click
    If TextBox7.Text = "A" Then
        TextBox8.Text = "M03"
        SerialPort1.WriteLine("M03")
        TextBox7.Text = "B"
        BtnSpn.Text = "Spindle OFF"
        SerialPort1.WriteLine("M03")
    Else
        TextBox8.Text = "M05"
        TextBox7.Text = "A"
        BtnSpn.Text = "Spindle ON"
        SerialPort1.WriteLine("M05")
    End If
End Sub

'Home
Private Sub BtnHome_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles BtnHome.Click
    TextBox11.Text = "G00 X0"
    TextBox10.Text = "G00 Y0"
    TextBox9.Text = "G00 Z0"
    SerialPort1.WriteLine("G00 X0 Y0 Z0")
End Sub

```

```
'exit
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    End
End Sub
```

```
'Stop
Private Sub BtnStop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnStop.Click
    TmrSend.Enabled = False
    TmrSend.Stop()
    TextBox1.Text = "M02"
    If TextBox1.Text = "M02" Then
        TextBox13.Text = "M02"
    End If
End Sub
```

```
'Refresh
Private Sub BtnRefresh_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnRefresh.Click
    System.Threading.Thread.Sleep(100)
    myPort = IO.Ports.SerialPort.GetPortNames()
    ComboBox2.Items.Add(9600)
    ComboBox2.Items.Add(19200)
    ComboBox2.Items.Add(38400)
    ComboBox2.Items.Add(57600)
    ComboBox2.Items.Add(115200)
    For i As Integer = 0 To UBound(myPort)
        ComboBox1.Items.Add(myPort(i))
    Next
    ComboBox1.Text = ComboBox1.Items.Item(0)
    ComboBox2.Text = ComboBox2.Items.Item(0)
    Button3.Enabled = False
    Button2.Enabled = False
    Button1.Enabled = False
End Sub
```

```
Private daftarPerintah As IList(Of String) = New List(Of String)
Private posisiPerintah As IList(Of Integer) = New List(Of Integer)
```

```

Private pathPerintah As IList(Of PointF) = New List(Of PointF)
Private sedangProses As Boolean = False
Private nomorPerintah As Integer = 0
Private warnaTap As Color = Color.Red
Private konstantaProyeksi As Single = 5.0F
Private intervalMilliDetik As Integer = 700
Protected Overrides Sub OnLoad(ByVal e As EventArgs)
    MyBase.OnLoad(e)
    tmrProses.Interval = intervalMilliDetik
    AddHandler tmrProses.Tick, AddressOf tmrProses_Tick
End Sub
Private Sub tmrProses_Tick(ByVal sender As Object, ByVal e As
EventArgs)
    If sedangProses Then Return
    sedangProses = True
    pnlBoard.Invalidate()
    ListBox1.SelectedIndex = posisiPerintah(nomorPerintah)
    nomorPerintah += 1
    If nomorPerintah >= daftarPerintah.Count() Then nomorPerintah =
0
    sedangProses = False
End Sub
Private Sub pnlBoard_Paint(ByVal sender As Object, ByVal e As
PaintEventArgs) Handles pnlBoard.Paint
    Try
        Dim g = e.Graphics
        Using _
            br As New SolidBrush(warnaTap),
            pn As New Pen(Color.DodgerBlue, 2) With {.LineJoin =
Drawing2D.LineJoin.Round}

'Dim rct = New RectangleF(pathPerintah(nomorPerintah), ukuranTap)
            If (nomorPerintah > 10) Then
                g.DrawLines(pn, pathPerintah.Take(nomorPerintah +
1).ToArray())
            End If
            pn.Width /= 4.0F : pn.Color = warnaTap
            g.DrawPolygon(pn, pathPerintah.Take(nomorPerintah +
1).ToArray())

```

```

        'g.FillEllipse(br, rct)
    End Using
Catch
End Try
End Sub

```

```

Private Function LokasiTapDariDaftarPerintah(ByVal v As String,
Optional ByVal separator As Char = " "c) As PointF
    Dim pts As New PointF
    Dim p = v.Split(separator)
    If p.Length > 3 AndAlso p(1).StartsWith("X") AndAlso
p(2).StartsWith("Y") Then
        Dim x = 10, y = 10
        If (
            Single.TryParse(p(1).Substring(1, p(1).Length - 4), x) AndAlso
            Single.TryParse(p(2).Substring(1, p(2).Length - 4), y)) Then

            pts = New PointF(x * 25, y * 25)
        End If
    End If
    Return pts
End Function

```

```

Private Sub BtnPause_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnPause.Click
    If BtnPause.Text = "Pause" Then
        BtnPause.Text = "Continue"
        TmrSend.Enabled = False
        TmrSend.Stop()

    Else
        BtnPause.Text = "Pause"
        Refresh()
        TextBox12.Text = String.Join(" ", ListBox1.Items.Cast(Of
String).Where(Function(w) w.StartsWith("G00") OrElse
w.StartsWith("G01") OrElse w.StartsWith("M03") OrElse
w.StartsWith("M05") OrElse w.StartsWith("M02") OrElse
w.StartsWith("G20") OrElse w.StartsWith("(") OrElse

```



```

w.StartsWith("G21")      OrElse      w.StartsWith("G90")      OrElse
w.StartsWith("G91")).ToArray())
    pnlBoard.CreateGraphics().Clear(pnlBoard.BackColor)
    TmrSend.Enabled = True
End If

End Sub
End Class

```

-----Halaman ini sengaja dikosongkan-----



RIWAYAT HIDUP

Nama : Aidin Amsyar
TTL : Lamongan, 12 Maret 1996
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat Rumah : Ds. Payaman RT.02/
RW.13, Kec. Solokuro,
Lamongan
Telp/HP : 085608050267
E-mail : amsyareden@gmail.com

RIWAYAT PENDIDIKAN

- 1999-2001 : TK Aisyiyah Bustanul Athfal
- 2001-2007 : Mi- Muhammadiyah 05 Palirangan
- 2007-2011 : MTs. N Model Babat
- 2011-2014 : SMA Negeri 1 Lamongan
- 2014-sekarang : Departemen Teknik Elektro Otomasi, ITS

PENGALAMAN KERJA

- Kerja Praktek di *Section of Packer&Port Elect&Instr Maint* PT Semen Indonesia (Persero)Tbk (2016)

PENGALAMAN ORGANISASI

- Kadep Peminjaman dan Pemeliharaan Alat SMASAPALA (2012-2013)
- Staff Departemen Syi'ar LDJ Salman Al-Farisi ITS (2015-2016)
- Kadep Departemen Syi'ar LDJ Salman Al-Farisi ITS (2016-2017)

-----Halaman ini sengaja dikosongkan-----